

Московский государственный университет им. М. В. Ломоносова  
Механико-математический факультет  
Кафедра математической логики и теории алгоритмов

Вступительный реферат в аспирантуру

# Экстракторы и эффективный вариант теоремы Мучника

Д.Мусатов

Научный руководитель: д. ф.-м. н., профессор Н. К. Верещагин

Москва  
2006

## Аннотация

В 1999 году Ан. Мучник доказал следующую теорему: для любых двоичных слов  $A$  и  $B$  найдется такое слово  $X$  длины примерно  $K(A|B)$ , что  $K(X|A) \approx 0$  и  $K(A|B, X) \approx 0$ .  $X$  выбирался как образ  $A$  под действием хеш-функции, выбранной из некоторого небольшого класса функций. Существование класса с нужными свойствами устанавливалось вероятностным методом. Ан. Мучник поставил вопрос: можно ли построить такое семейство хеш-функций, чтобы хеш-значение вычислялось полиномиальным алгоритмом по слову  $A$  и номеру функции? В работе показано, что ответ на этот вопрос положительный, если немного ослабить формулировку исходной теоремы. А именно, в качестве семейства хеш-функций надо взять экстрактор.

Экстрактором называется функция  $Ext: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ , такая что для любого распределения вероятностей на  $\{0, 1\}^n$  с большой минимальной энтропией и равномерного распределения на  $\{0, 1\}^d$  индуцированное распределение на  $\{0, 1\}^m$  близко к равномерному. Понятие экстрактора появилось в начале 1990-х годов и нашло много приложений в различных областях теории сложности. В литературе описаны различные конструкции экстракторов, обзор некоторых из них дан в настоящей работе. Явное построение экстрактора с оптимальными параметрами, существование которого установлено вероятностным методом, является открытой проблемой.

Применение экстракторов не только отвечает на вопрос, поставленный Ан. Мучником, но и является инструментом для доказательства аналогичных теорем для колмогоровской сложности с ограничением на ресурсы. В настоящей работе приведены первые результаты, полученные в этом направлении: теорема для сложности с полиномиальным ограничением на память и теорема для сложности с полиномиальным ограничением на время, где для декодирования используется алгоритм из класса АМ.

# 1. Введение

В этом разделе дана мотивировка основных понятий теории экстракторов и кратко описаны полученные результаты об условном кодировании. Все строгие определения и формулировки будут также даны в последующих разделах.

## 1.1. Понятие экстрактора

Решение многих задач в криптографии и других областях теории алгоритмов значительно упрощается за счет использования в алгоритмах случайных битов. При этом биты должны быть «действительно случайными», то есть независимыми и принимающими значения 0 и 1 с вероятностями  $1/2$ . Однако в природе такие идеальные распределения практически не встречаются. Например, если мы соорудим прибор, регистрирующий случайные явления, возникающие на квантовом уровне, априори ниоткуда не следует независимость битов, которые он будет выдавать. Тем не менее, хочется уметь применять вероятностные алгоритмы, имея в распоряжении только такие «квазислучайные» биты. Возникает вопрос: нельзя ли алгоритмически «извлечь случайность» из «квазислучайного» распределения и получить таким образом некоторое меньшее число «действительно случайных» или хотя бы «почти случайных» битов? Чтобы ответить на этот вопрос, нужно сначала формализовать понятие «квазислучайности». Наиболее естественным представляется следующее определение: будем говорить, что распределение «содержит  $k$  случайных битов», если его *минимальная энтропия* не меньше  $k$ , т. е. вероятность любого элемента не превышает  $2^{-k}$ .

Пусть дано распределение на  $\{0, 1\}^n$  с минимальной энтропией  $n - 1$ . Можем ли мы детерминированным алгоритмом извлечь хотя бы один «действительно случайный» бит? Очевидно, нет! Ведь хотя бы для одного бита  $b \in \{0, 1\}$  прообраз  $b$  будет содержать больше  $2^{n-1}$  слов, и если мы возьмем равномерное распределение на этом прообразе, то алгоритм заведомо выдаст  $b$ , и никакой случайности не будет.

Поэтому нам нужно ослабить требование. А именно, позволим нашему «извлекающему случайность» алгоритму использовать некоторое небольшое число  $d$  («действительно») случайных битов. Кроме того, ослабим требование к выходу алгоритма: будем требовать только, чтобы «извлеченные» биты были «почти случайными», то есть чтобы вероятность любого множества слов отличалась от его доли не больше, чем на какое-то маленькое число  $\varepsilon$ . Такие ослабленные требования приводят нас к классическому определению экстрактора:

**Определение 1.1.** Функция  $Ext: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  называется  $(k, \varepsilon)$ -экстрактором, если для любого распределения  $X$  на  $\{0, 1\}^n$  с минимальной энтропией не меньше  $k$  и равномерного распределения  $U$  на  $\{0, 1\}^d$  индуцированное распределение  $Ext(X, U)$   $\varepsilon$ -близко к равномерному на  $\{0, 1\}^m$ .

В некоторых приложениях не нужно, чтобы индуцированное распределение было близко к равномерному, а нужно лишь, чтобы почти все элементы в образе имели ненулевую вероятность. В таком случае функцию называют *дисперсером*. Часто экстрак-

торы и дисперсеры рассматривают не как функции, а как двудольные графы: левая доля отождествляется с  $\{0, 1\}^n$ , правая — с  $\{0, 1\}^m$ , а множество рёбер, выходящих из фиксированной вершины левой доли, — с  $\{0, 1\}^d$ . Более подробно понятие дисперсера и эквивалентность функций и графов рассмотрены в разделе 2.2.

Однако встает вопрос: что мы можем получить от такого определения? Ведь нам все равно нужны «действительно случайные» биты, пусть и в меньшем количестве. Выясняется, что существуют экстракторы, для которых  $d = O(\log n)$ , и этого вполне достаточно для некоторых приложений. Пусть, например, мы хотим симулировать некоторый алгоритм  $A \in \text{BPP}$ , использующий  $m$  случайных битов, при помощи  $n$  «квазислучайных битов» и экстрактора с подходящими параметрами. Можно показать, что с высокой вероятностью для большинства слов  $y \in \{0, 1\}^d$  алгоритм  $A$ , получивший вместо случайных битов биты, выданные экстрактором, выдаст правильный ответ. А это значит, что мы можем *перебрать* все  $2^d = \text{poly}(n)$  вариантов слова  $y$ , для каждого из них запустить экстрактор, передать полученные биты алгоритму  $A$  и выбрать наиболее часто встретившийся ответ.

Чтобы использовать экстрактор для симуляции алгоритмов, работающих полиномиальное время, нужно чтобы сам экстрактор мог быть вычислен за полиномиальное время. Вероятностным методом можно доказать существование для любых  $k < n$  экстрактора с  $d = \log n + O(1)$  и  $m = k + d - O(1)$ . В статье [11] доказано, что это оптимальные параметры, то есть не может быть экстрактора с меньшим  $d$  или большим  $k$ . Однако ни одна из известных конструкций не достигает этих параметров. Все они имеют либо большее  $d$ , либо меньшее  $m$ , либо работают не для всех значений  $k$ .

## 1.2. Понятие колмогоровской сложности

Пусть  $A$  и  $B$  — произвольные двоичные слова. Условной колмогоровской сложностью  $K(A|B)$  слова  $A$  относительно слова  $B$  называется минимальная длина программы, переводящей  $B$  в  $A$ . Колмогоровской сложностью  $K(A)$  слова  $A$  называется  $K(A|\Lambda)$ , где  $\Lambda$  — пустое слово, т. е. длина минимальной программы, порождающей слово  $A$ . При этом, разумеется, колмогоровская сложность зависит от способа задания программ, однако различные естественные определения дают отличие сложности не более, чем на  $O(\log n)$ , для слов длины  $n$ . В формулировке основной теоремы мы пренебрегаем таким различием, поэтому нам не нужно уточнять определение.

Также рассматривают колмогоровскую сложность с ограничением на ресурсы. Сложностью  $C^{t,s}(A|B)$  называется минимальная длина программы, переводящей  $B$  в  $A$  и работающей время  $t$  на зоне  $s$ . В этом случае способ задания уже имеет значение, более того, имеет значение и используемая модель вычислений. Так, различают сложности  $C$  для детерминированных алгоритмов,  $CN$  для недетерминированных,  $CBP$  для вероятностных и другие. Все необходимые точные определения даны в разделе 8.1.

### 1.3. Условное кодирование и теорема Мучника

Рассмотрим такую задачу алгоритмической теории передачи информации. Пусть в точке  $E$  известно некоторое слово  $A$ . Точку  $E$  соединяет с точкой  $D$  канал ограниченной пропускной способности  $k$ , через который нужно передать некоторое сообщение  $X$ . В точке  $D$  известно некоторое другое слово  $B$ , и нужно по переданному сообщению  $X$  расшифровать слово  $A$ . При этом операции зашифровки и расшифровки должны выполняться алгоритмически. Вопрос: при каких условиях на  $A$  и  $B$  такое сообщение  $X$  найдется? Очевидное необходимое условие состоит в том, что  $k \geq K(A|B)$ . Неожиданный результат, установленный Ан. Мучником в работе [6] состоит в том, что это (с определенными оговорками) и достаточное условие. Это кажется удивительным: ведь на этапе кодирования мы еще не знаем слова  $B$ , тем не менее нам почти не нужно дополнительной информации для условного кодирования  $A$ . А именно, выполнена следующая

**Теорема 1.1** ([6]). Пусть  $A$  и  $B$  — двоичные слова длины не более  $n$ . Тогда найдется такое слово  $X$  длины не более  $K(A|B) + O(\log n)$ , что  $K(X|A) \leq O(\log n)$  и  $K(A|B, X) \leq O(\log n)$ .

Эту теорему можно сформулировать и так: среди всех программ, задающих  $A$  при известном  $B$ , найдется простая относительно  $A$ .

*Идея доказательства.* Идея доказательства состоит в том, чтобы рассмотреть семейство хеш-функций, сопоставляющих словам длины  $n$  слова длины  $m = K(A|B)$ , и взять в качестве  $X$  образ  $A$  под действием одной из этих функций. Это семейство должно обладать тем свойством, что для каждого слова  $A$  найдется хеш-функция, такая что по ее значению  $X$  и слову  $B$  можно легко (т.е. с логарифмическим числом дополнительной информации) восстановить  $A$ . Кроме того, это семейство должно быть полиномиального размера, чтобы  $X$  можно было задать номером соответствующей функции. Существование таких семейств хеш-функций устанавливается вероятностным методом.  $\square$

Докладывая этот результат на Колмогоровском семинаре в Московском университете в 1999 году, Ан. Мучник поставил вопрос: возможно ли построить *полиномиально вычислимое* семейство хеш-функций с указанным свойством, т.е. можно ли вычислить хеш-значение по слову  $A$  и номеру функции за полиномиальное время? Настоящая работа частично отвечает на этот вопрос: да, можно, если в условии теоремы заменить поправки  $O(\log n)$  на  $\text{polylog}(n)$ , при этом степень логарифма может быть сделана равной  $2 + \varepsilon$  для любого  $\varepsilon > 0$ . А именно, оказывается, что вместо хеш-функций, определенных Мучником, можно использовать экстракторы. Если будут построены оптимальные экстракторы, вычисляемые за полиномиальное время, то это сразу даст полный ответ на вопрос Мучника.

Применение экстракторов также позволяет доказать аналогичный результат для кодирования с несколькими условиями. Кроме того, данная техника позволяет получать похожие теоремы для колмогоровской сложности с ограничением на ресурсы. В работе приведены первые результаты, полученные в этом направлении:

**Теорема 1.2.** *Найдётся такой полином  $p_0 = p_0(n)$ , что для любого полинома  $p \geq p_0$  найдутся полином  $q$  и  $t$ , такие что для любых слов  $A$  и  $B$  длины не более  $n$ , таких что  $C^{\infty,p}(A|B) \leq k$ , найдётся слово  $X$  длины не более  $k + O(1)$ , такое что  $C^{t,q}(X|A) \leq O(\log^3 n)$  и  $C^{\infty,q}(A|B, X) \leq O(\log^3 n)$ .*

**Теорема 1.3.** *Для всякого полинома  $p$  найдётся полином  $q$ , такой что для произвольных слов  $A$  и  $B$  длины не более  $n$ , таких что  $C^{p,\infty}(A|B) \leq k$ , найдётся слово  $X$  длины не более  $k + O(\log^3 n)$ , такое что  $C^{q,\infty}(X|A) \leq O(\log^3 n)$  и  $CAM^{q,\infty}(A|B, X) \leq O(\log n)$ .*

Определения использованных сложностей даны в разделе 8.1, доказательства теорем — в разделах 8.4 и 8.5.

## 1.4. Организация дальнейшего текста

В разделе 2 даны определения экстрактора и доказаны простейшие факты. В разделе 3 вероятностным методом доказано существование экстракторов с оптимальными параметрами. В разделах 4–7 с разной степенью подробности приведены некоторые конструкции полиномиально вычислимых экстракторов. В разделе 8 при помощи экстракторов доказаны теорема Мучника и ещё эффективные варианты.

# 2. Основные определения и понятия теории экстракторов

## 2.1. Минимальная энтропия и статистическое расстояние

Для начала дадим два вспомогательных определения. Будем рассматривать вероятностные распределения на конечных множествах. Вероятность элемента  $a$  в распределении  $X$  будем обозначать как  $X(a)$ , вероятность множества  $S$  — как  $X(S)$ .

**Определение 2.1.** Пусть дано вероятностное распределение  $X$  на конечном множестве  $A$ . Тогда его минимальной энтропией называется величина

$$H_{\infty}(X) = \min_{a \in A} (-\log_2(X(a))).$$

Таким образом, если  $H_{\infty}(X) > k$ , то все элементы множества  $A$  имеют вероятности меньше  $2^{-k}$ .

**Определение 2.2.** Пусть даны вероятностные распределения  $X$  и  $Y$  на конечном множестве  $A$ . Тогда статистическим расстоянием между ними называется величина

$$\text{dist}(X, Y) = \frac{1}{2} \|X - Y\| = \frac{1}{2} \sum_{a \in A} |X(a) - Y(a)| = \max_{S \subseteq A} |X(S) - Y(S)|.$$

Будем говорить, что распределение  $X$   $\varepsilon$ -близко к  $Y$ , если  $\text{dist}(X, Y) \leq \varepsilon$ , иными словами, вероятности каждого события по этим распределениям отличаются не более чем на  $\varepsilon$ .

## 2.2. Экстракторы и дисперсеры

Как уже отмечалось, экстракторы и дисперсеры можно рассматривать как функции и как двудольные графы. Приведем формальные определения.

Пусть  $G$  — двудольный граф (возможно, с кратными ребрами) с  $N$  вершинами в левой доле,  $M$  вершинами в правой доле и степенью  $D$  каждой вершины из левой доли. Обозначим через  $[N] = \{1, \dots, N\}$  множество вершин левой доли, через  $[M] = \{1, \dots, M\}$  — правой, а через  $E$  множество ребер.

Пусть  $\Gamma(a) = \{z \in [M] \mid (a, z) \in E\}$  — множество всех соседей вершины  $a$  из левой доли,  $\Gamma(A) = \bigcup_{a \in A} \Gamma(a)$  — множество всех соседей подмножества  $A$  левой доли.

**Определение 2.3.** Граф  $G = ([N], [M], E)$  называется  $(K, \varepsilon)$ -дисперсером, если  $\forall A \subset [N]$ ,  $|A| \geq K$ ,  $|\Gamma(A)| \geq (1 - \varepsilon)M$ .

Это определение объясняет выбор названия «дисперсер»: все достаточно большие подмножества левой доли имеют в соседях почти все вершины правой доли, т. е. граф «рассеивает» (англ. disperse - рассеивать) их по правой доле.

**Определение 2.4.** Граф  $G = ([N], [M], E)$  называется  $(K, \varepsilon)$ -экстрактором, если  $\forall A \subset [N]$ ,  $|A| \geq K$ ,  $\forall B \subset [M]$

$$\left| \frac{|E(A, B)|}{|A| \cdot D} - \frac{|B|}{M} \right| < \varepsilon,$$

где  $E(A, B)$  — множество ребер, идущих из множества  $A$  в множество  $B$  в графе  $G$ .

Нетрудно заметить, что  $(K, \varepsilon)$ -экстрактор также является  $(K, \varepsilon)$ -дисперсером, достаточно взять  $B = \Gamma(A)$ .

Теперь дадим определение экстракторов и дисперсеров в терминах функций. Обозначим  $\{0, 1\}^l$  через  $(l)$ , а равномерное распределение на этом множестве через  $U_l$ . Будем рассматривать функции  $F : (n) \times (d) \rightarrow (m)$ .

**Определение 2.5.** Функция  $F : (n) \times (d) \rightarrow (m)$  называется  $(k, \varepsilon)$ -дисперсером, если для всех вероятностных распределений  $X$  на  $\{0, 1\}^n$  с  $H_\infty(X) \geq k$  и множеств  $B \subset \{0, 1\}^m$ ,  $|B| \geq (1 - \varepsilon)M$

$$\Pr [F(X, U_d) \in B] > 0.$$

**Определение 2.6.** Функция  $F : (n) \times (d) \rightarrow (m)$  называется  $(k, \varepsilon)$ -экстрактором, если для всех вероятностных распределений  $X$  на  $\{0, 1\}^n$  с  $H_\infty(X) \geq k$

$$\text{dist}(F(X, U_d), U_m) < \varepsilon.$$

При таком определении ясен смысл названия «экстрактор». Функция  $G$  «извлекает» (англ. extract - извлекать)  $m$  «почти случайных» битов из поданных ей на вход  $n$  «квазислучайных» битов при помощи  $d$  «действительно случайных» битов.

Заметим, что если функция является экстрактором, то любой ее префикс является экстрактором с теми же параметрами  $k$  и  $\varepsilon$ . Однако в теореме Мучника полезно, чтобы префиксы являлись экстракторами с лучшими параметрами. Дадим следующее формальное определение.

**Определение 2.7.** Обозначим через  $X|_q$  префикс слова  $X$  длины  $q$ . Функцию  $F : (n) \times (d) \rightarrow (m)$  назовем префиксным  $(k, \varepsilon)$ -экстрактором<sup>1</sup>, если для всех  $i = 0, \dots, k$  функция  $F|_{m-i} : (n) \times (d) \rightarrow (m-i)$ , определенная равенством  $F|_{m-i}(u, y) = (F(u, y))|_{m-i}$ , является  $(k-i, \varepsilon)$ -экстрактором.

Функции на словах и двудольные графы описанного вида естественным образом соответствуют друг другу при  $N = 2^n$ ,  $M = 2^m$  и  $D = 2^d$ . Левая доля графа отождествляется со словами длины  $n$ , правая — со словами длины  $m$ , а рёбра, проведенные из фиксированной вершины левой доли, — со словами длины  $d$ . При этом нумерация рёбер произвольна, т.е. разным функциям может соответствовать один и тот же граф. Для префиксного экстрактора тоже можно и полезно рассмотреть соответствующий ему граф, но он не будет иметь естественного определения в терминах графа, поскольку любое такое определение будет зависеть от нумерации вершин правой доли. Кроме того, в отличие от обычного экстрактора определение префиксного не обобщается на  $N, M, D$  и  $K$ , не являющиеся степенями двойки.

**Утверждение 2.1.** Обозначим граф, построенный по функции  $F$ , через  $G_F$ . Тогда

- (a)  $F$  является  $(k, \varepsilon)$ -дисперсером  $\Leftrightarrow G_F$  является  $(2^k, \varepsilon)$ -дисперсером.
- (b)  $F$  является  $(k, \varepsilon)$ -экстрактором  $\Leftrightarrow G_F$  является  $(2^k, \varepsilon)$ -экстрактором.

*Доказательство.* Переход от функций к графам очевиден: достаточно взять в качестве распределения  $X$  равномерное на  $A$ .

Обратный переход следует из того, что любое распределение с  $H_\infty(X) \geq k$  представляется в виде взвешенной суммы равномерных на множествах размера  $K$ . Будем следовать доказательству этого факта, предложенному М.Бабенко, а именно докажем по индукции следующее эквивалентное утверждение: Пусть дано  $N$  неотрицательных чисел, каждое из которых не превышает  $1/K$  общей суммы. Назовем «операцией» одновременное уменьшение  $K$  чисел на одну и ту же величину («величину операции»). Тогда не более чем за  $N$  операций можно обратить все числа в 0.

Выберем какие-нибудь  $K$  ненулевых чисел. Уменьшим их на максимально возможную величину, так чтобы не нарушить условие. Тогда либо одно из них обратится в нуль, и мы можем применить предположение индукции для  $N-1$  и  $K$  непосредственно, либо одно из невыбранных чисел ( $x$ ) станет равным  $1/K$  общей суммы. В таком случае каждое из остальных чисел не превышает  $1/(K-1)$  суммы остальных чисел, и мы можем для остальных чисел применить предположение индукции для  $N-1$  и  $K-1$ . А именно, добавим к каждому набору из предположения индукции число  $x$  и таким образом вместе с исходной операцией получим последовательность операций для исходного набора. Действительно, после первой операции сумма всех чисел, кроме  $x$ , равна  $(K-1)/K$  от суммы всех чисел. Значит, сумма всех величин операций из предположения индукции равна  $1/(K-1) \cdot (K-1)/K = 1/K$  от общей суммы, то есть как раз  $x$ , что и нужно.  $\square$

---

<sup>1</sup>Насколько известно автору, понятие префиксного экстрактора ранее не встречалось в литературе, однако соответствующее свойство неоднократно отмечалось различными авторами.



### 2.3. Полиномиально вычислимые экстракторы и постановка задачи оптимизации параметров

Вероятностным методом можно доказать, что для всех  $n, k$  и  $\varepsilon$  существуют экстракторы с  $d = \log(n - k) + 2 \log(1/\varepsilon) + O(1)$  и  $m = k + d - \log(1/\varepsilon) - O(1)$ . В работе [11] также показано, что это оптимальные параметры. Однако в большинстве приложений необходимо, чтобы экстракторы были вычислимы за полиномиальное время. Определим формально, что это значит.

**Определение 2.8.** Пусть для функций  $k(n), \varepsilon(n), d(n), m(n)$  задано семейство  $Ext = \{Ext_n\}$  отображений  $Ext_n: \{0, 1\}^n \times \{0, 1\}^{d(n)} \rightarrow \{0, 1\}^{m(n)}$ . Тогда оно называется полиномиально вычислимым  $(k, \varepsilon)$ -экстрактором, если  $Ext_n$  является  $(k(n), \varepsilon(n))$ -экстрактором при всех  $n$ , и  $Ext_n$  вычислима за полиномиальное от длины входа время.

До сих пор неизвестно, можно ли построить полиномиально вычислимые экстракторы с оптимальными параметрами. Поэтому задача оптимизации ставится так: либо при всех  $k$  и  $\varepsilon$  достичь наибольшего возможного  $m$  для оптимального  $d$ , либо достичь наименьшего возможного  $d$  для оптимального  $m$ , либо достичь оптимальных  $m$  и  $d$  для некоторых значений  $k$ .

## 3. Вероятностное доказательство существования экстракторов

Покажем, что случайный граф с определенными параметрами с положительной вероятностью является экстрактором.

**Теорема 3.1** ([11]). *Для всех  $1 < K \leq N$ ,  $M > 0$  и  $\varepsilon > 0$  существуют:*

- (a)  $(K, \varepsilon)$ -дисперсер для  $D = \lceil \frac{M}{K} (\ln \frac{1}{\varepsilon} + 1) + \frac{1}{\varepsilon} (\ln \frac{N}{K} + 1) \rceil$ ;
- (b)  $(K, \varepsilon)$ -экстрактор для  $D = \lceil \max \{ \frac{M}{K} \cdot \frac{\ln 2}{\varepsilon^2}, \frac{1}{\varepsilon^2} (\ln \frac{N}{K} + 1) \} \rceil$ ;
- (c) при условии, что  $N = 2^n$ ,  $M$  и  $K$  суть степени двойки, префиксный  $(K, \varepsilon)$ -экстрактор для  $\log D = \lceil \log (\max \{ \frac{M}{K} \cdot \frac{\ln 2}{\varepsilon^2}, \frac{1}{\varepsilon^2} (1 + \ln 2 + \ln N) \}) \rceil$ .<sup>2</sup>

Прежде чем доказывать теорему, переформулируем ее в терминах функций:

**Теорема 3.2.** *Для всех  $1 \leq k \leq n$  и  $\varepsilon > 0$  существуют:*

- (a)  $(k, \varepsilon)$ -дисперсер для  $d = \log(n - k) + \log(1/\varepsilon) + O(1)$  и  $m = k + d - \log \log(1/\varepsilon) - O(1)$ ;
- (b)  $(k, \varepsilon)$ -экстрактор для  $d = \log(n - k) + 2 \log(1/\varepsilon) + O(1)$  и  $m = k + d - 2 \log(1/\varepsilon) - O(1)$ ;
- (c) Префиксный  $(k, \varepsilon)$ -экстрактор для  $d = \log n + 2 \log(1/\varepsilon) + O(1)$  и  $m = k + d - 2 \log(1/\varepsilon) - O(1)$ .

---

<sup>2</sup>Этого пункта теоремы не было в работе [11].

В работе [11] доказаны следующие нижние оценки:

**Теорема 3.3** ([11]). (a) Если функция  $F: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  является  $(k, \varepsilon)$ -дисперсером, то  $d \geq \log(n - k) + \log(1/\varepsilon) - O(1)$  и  $d + k - m \geq \log \log(1/\varepsilon) - O(1)$ .

(b) Если функция  $F: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  является  $(k, \varepsilon)$ -экстрактором, то  $d \geq \log(n - k) + 2 \log(1/\varepsilon) - O(1)$  и  $d + k - m \geq 2 \log(1/\varepsilon) - O(1)$ .

Таким образом, параметры, достигнутые в теореме 3.2, являются оптимальными.

Из оценки для экстракторов следует оценка для префиксных экстракторов:  $d \geq \log n + 2 \log(1/\varepsilon) - O(1)$ . Таким образом, заявленные параметры являются оптимальными.

*Доказательство теоремы 3.1.* Докажем вначале оценку для дисперсеров. Возьмем двудольный граф  $G = ([N], [M], E)$  с  $N$  вершинами в левой доле,  $M$  вершинами в правой, в котором из каждой вершины левой доли случайным образом выпущено  $D$  рёбер. Положим  $L = \lceil \varepsilon M \rceil$ . Чтобы этот граф не был  $(K, \varepsilon)$ -дисперсером, должны найтись подмножества размера  $L$  в правой доле и размера  $K$  в левой, которые не соединяет ни одно ребро. Таким образом,  $\Pr[G \text{ не является } (K, \varepsilon)\text{-дисперсером}]$  не превосходит

$$C_N^K \cdot C_M^L \cdot \left(1 - \frac{L}{M}\right)^{KD} < \left(\frac{eN}{K}\right)^K \left(\frac{eM}{L}\right)^L \exp\left(-\frac{LKD}{M}\right). \quad (1)$$

В этом переходе мы воспользовались неравенствами  $C_n^k < \left(e \frac{n}{k}\right)^k$ , которое легко доказать по индукции, и  $1 - x \leq e^{-x}$ . Подставив  $D$  из условия теоремы, получаем

$$\begin{aligned} \exp\left(\frac{LKD}{M}\right) &\geq \exp\left(L\left(\ln \frac{1}{\varepsilon} + 1\right) + K \cdot \frac{L}{M} \cdot \frac{1}{\varepsilon} \left(\ln \frac{N}{K} + 1\right)\right) \geq \\ &\geq \exp\left(L \ln \frac{eM}{L} + K \ln \frac{eN}{K}\right) = \left(\frac{eN}{K}\right)^K \left(\frac{eM}{L}\right)^L, \end{aligned} \quad (2)$$

откуда правая часть (1) не превосходит 1. Значит, случайный граф с указанными параметрами не является  $(K, \varepsilon)$ -дисперсером с вероятностью меньше 1, т.е. является  $(K, \varepsilon)$ -дисперсером с положительной вероятностью, что и требовалось доказать.

Теперь докажем оценку для экстракторов. Заметим вначале, что изначальное требование, чтобы  $\forall A \subset [N], |A| \geq K \forall B \subset [M]$  выполнялось

$$\left| \frac{|E(A, B)|}{|A| \cdot D} - \frac{|B|}{M} \right| < \varepsilon,$$

можно заменить на более слабое:  $\forall A \subset [N], |A| = K \forall B \subset [M] |E(A, B)| < KD \left(\frac{|B|}{M} + \varepsilon\right)$ . Действительно, сведение к множествам размера ровно  $K$  следует из доказательства утверждения 2.1. Далее, пусть найдутся  $A \subset [N]$  и  $B \subset [M]$ , такие что  $|E(A, B)| \leq$

$KD \left( \frac{|B|}{M} - \varepsilon \right)$ . Тогда положим  $\overline{B} = [M] \setminus B$  и получим  $|E(A, \overline{B})| \geq KD \left( \frac{|B|}{M} + \varepsilon \right)$ , что противоречит ослабленному требованию.

Далее, возьмем вновь двудольный граф  $G = ([N], [M], E)$  с  $N$  вершинами в левой доле,  $M$  вершинами в правой, в котором из каждой вершины левой доли случайным образом выпущено  $D$  рѣбер. Фиксируем  $A \subset [N]$ ,  $|A| = K$  и  $B \subset [M]$ . Положим  $p = \frac{|B|}{M}$ . Оценим вероятность того, что  $|E(A, B)| \geq KD(p + \varepsilon)$ . Количество рѣбер, идущих из  $A$  в  $B$ , есть сумма  $KD$  независимых одинаково распределѣнных случайных величин, принимающих значение 1 с вероятностью  $p$  и значение 0 с вероятностью  $1 - p$ . По неравенству Чернова мы можем оценить

$$\Pr [|E(A, B)| \geq KD(p + \varepsilon)] \leq \exp(-2\varepsilon^2 KD).$$

Таким образом,  $\Pr [G \text{ не является } (K, \varepsilon)\text{-экстрактором}]$  не превосходит

$$\begin{aligned} C_N^K \cdot 2^M \exp(-2\varepsilon^2 KD) &< \left( \frac{eN}{K} \right)^K 2^M \exp(-2\varepsilon^2 KD) = \\ &= \left( e^{K(1+\ln(N/K))} \cdot e^{-\varepsilon^2 KD} \right) \cdot \left( e^{M \ln 2} \cdot e^{-\varepsilon^2 KD} \right). \end{aligned}$$

Поскольку  $D \geq \frac{1}{\varepsilon^2}(1 + \ln(N/K))$ , первый множитель не превосходит 1. Аналогично, поскольку  $D \geq \frac{M \ln 2}{\varepsilon^2 K}$ , второй множитель не превосходит 1. В итоге получаем, что

$$\Pr [G \text{ не является } (K, \varepsilon)\text{-экстрактором}] < 1,$$

что и требовалось доказать.

Перейдем к префиксным экстракторам. Вначале переведем определение на язык графов. Назовем блоком уровня  $i$  подмножество вершин правой доли размера  $2^i$ , которое состоит из всех слов, имеющих фиксированный префикс длины  $m - i$ . Будем говорить, что множество  $B$   $i$ -вложено в  $[M]$  ( $B \overset{i}{\subset} [M]$ ), если любой блок уровня  $i$  либо полностью входит в  $B$ , либо полностью не входит. Таким образом,  $i$ -вложенные подмножества естественным образом соответствуют множествам слов длины  $m - i$ . Теперь можно определить префиксный экстрактор так:  $\forall i = 0, \dots, k \ \forall A \subset [N], |A| = K/2^i \ \forall B \overset{i}{\subset} [M] \ |E(A, B)| < \frac{KD}{2^i} \left( \frac{|B|}{M} + \varepsilon \right)$ .

Следуя схеме доказательства для экстракторов, получим, что

$$\begin{aligned} \Pr [G \text{ не является } (K, \varepsilon)\text{-префиксным экстрактором}] &\leq \\ &\leq \sum_{i=0}^k C_N^{K/2^i} \cdot 2^{M/2^i} \exp(-2\varepsilon^2 KD/2^i) < \sum_{i=0}^k \left( \frac{eN}{K/2^i} \right)^{K/2^i} 2^{M/2^i} \exp(-2\varepsilon^2 KD/2^i) = \\ &= \sum_{i=0}^k \left( e^{K/2^i(1+\ln(2^i N/K))} \cdot e^{-\varepsilon^2 KD/2^i} \right) \cdot \left( e^{M \ln 2/2^i} \cdot e^{-\varepsilon^2 KD/2^i} \right) \leq \\ &\leq \sum_{i=0}^k \cdot \left( e^{K/2^i(1+\ln N - \varepsilon^2 D)} \right) \cdot \left( e^{(M \ln 2 - \varepsilon^2 KD)/2^i} \right). \end{aligned}$$

Второй множитель в каждом слагаемом, как и прежде, не превосходит 1. Поскольку  $D \geq \frac{1}{\varepsilon^2}(1 + \ln 2 + \ln N)$ , первый множитель в  $i$ -ом слагаемом не превосходит  $(1/2)^{K/2^i}$ . Значит, сумма всех первых множителей меньше 1, значит и вся сумма меньше 1, следовательно, исходная вероятность также меньше 1. Значит, граф с искомыми параметрами найдется.  $\square$

## 4. Экстракторы на базе хеш-функций

### 4.1. Базовая конструкция

**Определение 4.1.** Набор функций  $H = \{h : [N] \rightarrow [L]\}$  называется семейством хеш-функций со степенью коллизий  $\delta$ , если  $\forall x_1 \neq x_2 \in [N] \Pr_{h \in H} [h(x_1) = h(x_2)] \leq (1 + \delta)/L$ .

Пусть функции  $h \in H$  занумерованы произвольным образом. Тогда по семейству  $H$  можно построить экстрактор по формуле  $F(x, h) = h \cdot h(x)$ , где  $\cdot$  обозначает конкатенацию. Таким образом,  $D = |H|$ ,  $M = DL$ .

**Лемма 4.1** ([4]). Пусть  $H$  — семейство хеш-функций с вероятностью коллизий  $\delta$ . Тогда экстрактор, построенный по  $H$ , имеет параметры  $K = 2^k = O(L/\delta)$  и  $\varepsilon = O(\sqrt{\delta})$ .

*Доказательство.* Определим вероятность коллизий для распределения  $X$  как  $\text{col}(X) = \sum_a (X(a))^2$  — вероятность того, что независимо выбранные в соответствии с  $X$   $x_1$  и  $x_2$  совпадут. Покажем, что  $\text{col}(X) \leq \frac{1}{K}$  при  $H_\infty \geq k$ . Действительно,

$$\text{col}(X) = \sum_a (X(a))^2 \leq 2^{-k} \sum_a X(a) = 2^{-k} = \frac{1}{K}.$$

Далее, обозначим через  $Z$  распределение  $h \cdot h(x)$ , где  $h$  распределено равномерно, а  $x$  в соответствии с  $X$ , и посчитаем  $\text{col}(Z)$ .  $\text{col}(Z)$  — вероятность того, что для независимо выбранных  $(h_1, x_1)$  и  $(h_2, x_2)$  верно  $h_1 = h_2$  и  $h_1(x_1) = h_2(x_2)$ . Она равна домноженной на  $1/|H|$  вероятности того, что для случайной  $h \in H$  и  $x_1, x_2$ , выбранных в соответствии с  $X$ ,  $h(x_1) = h(x_2)$ . Вероятность того, что  $x_1 = x_2$ , равна  $\text{col}(X)$ . Если же  $x_1 \neq x_2$ , то эта вероятность не превосходит  $(1 + \delta)/L$ . В итоге получаем

$$\text{col}(Z) \leq \frac{1}{|H|} \left( \text{col}(X) + \frac{1 + \delta}{L} \right) \leq \frac{1}{M} + \frac{1}{|H|} \left( \frac{1}{K} + \frac{\delta}{L} \right). \quad (3)$$

Далее,

$$\text{col}(Z) = \sum_a (Z(a))^2 = \sum_a \left( Z(a) - \frac{1}{M} \right)^2 + \sum_a \frac{2Z(a)}{M} - \sum_a \frac{1}{M^2} = \sum_a \left( Z(a) - \frac{1}{M} \right)^2 + \frac{1}{M},$$

откуда с учетом (3) получаем  $\sum_a (Z(a) - 1/M)^2 \leq (1/K + \delta/L)/|H|$ , откуда

$$\sum_a \left| Z(a) - \frac{1}{M} \right| \leq \sqrt{M \left( \frac{1}{K} + \frac{\delta}{L} \right) \frac{1}{|H|}} = \sqrt{\frac{L}{K}} + \delta,$$

что при  $K = L/\delta$  влечет  $O(\sqrt{\delta})$ -близость распределения  $Z$  к равномерному, что и требовалось доказать.  $\square$

В [15] доказана следующая

**Лемма 4.2.** *Для всех  $1 \leq L \leq N$  и  $\varepsilon > 0$  можно построить семейство  $H$  хеш-функций, отображающих  $[N]$  в  $[L]$ , с вероятностью коллизий  $\varepsilon$  размера  $|H| = \text{poly}(n, \varepsilon^{-1}, L)$ .*

Отсюда выводим

**Следствие 4.3.** *Для всех  $M \leq N$  и  $\varepsilon > 0$  можно построить  $(k, \varepsilon)$ -экстрактор с  $D = \text{poly}(n, \varepsilon^{-1}, M)$  и  $k = m - d + O(\log \varepsilon^{-1})$ .*

Заметим, что построенный экстрактор требует, вообще говоря, очень много случайных битов ( $d$  полиномиально зависит от  $m$ ), однако при малых  $m$  (точнее, при  $m = \text{polylog}(n)$ ) построенный экстрактор является оптимальным.

## 4.2. Композиция экстракторов

**Определение 4.2.** Пусть  $F_1 : (n_1) \times (d_1) \rightarrow (m_1)$  и  $F_2 : (n_2) \times (d_2) \rightarrow (d_1)$  суть экстракторы. Тогда определим их композицию  $F_1 \circ F_2 : (n_1 + n_2) \times (d_2) \rightarrow (m_1)$  по формуле  $F_1 \circ F_2(x_1, x_2, y) = F_1(x_1, F_2(x_2, y))$ .

Таким образом, почти случайные биты, полученные с помощью второго экстрактора, направляются на вход первому экстрактору в качестве случайных. Разумеется, свойства первого экстрактора при этом могут ухудшиться. Однако, если на вход композиции подать не произвольное распределение  $X$ , а распределение специального вида, то свойства сохранятся.

Дадим формальное определение:

**Определение 4.3.** Пусть  $X_1$  и  $X_2$  суть (определенные на одном вероятностном пространстве) случайные величины, принимающие значения на  $\{0, 1\}^{n_1}$  и  $\{0, 1\}^{n_2}$  соответственно. Будем говорить, что они образуют  $(k_1, k_2)$ -блочный источник, если

1.  $H_\infty(X_1) \geq k_1$ .
2. При любом фиксированном  $x_1$   $H_\infty(X_2|X_1 = x_1) \geq k_2$ .

Это определение обобщается на произвольное число случайных величин.

Имеет место несложная

**Лемма 4.4.** *Пусть  $F_1 : (n_1) \times (d_1) \rightarrow (m_1)$  —  $(k_1, \varepsilon_1)$ -экстрактор, а  $F_2 : (n_2) \times (d_2) \rightarrow (d_1)$  —  $(k_2, \varepsilon_2)$ -экстрактор. Пусть также  $(X_1, X_2)$  —  $(k_1, k_2)$ -блочный источник. Тогда распределение  $F_1 \circ F_2(X_1, X_2, U_{d_2})$   $(\varepsilon_1 + \varepsilon_2)$ -близко к равномерному.*

*Доказательство.* Обозначим через  $W$  случайную величину  $F_2(X_2, U_{d_2})$ . Зафиксируем значение  $x_1$ , тогда при условии  $X_1 = x_1$  распределение  $W$   $\varepsilon_2$ -близко к равномерному. Значит, распределение пары  $(X_1, W)$   $\varepsilon_2$ -близко к распределению  $(X_1, U_{d_1})$ . Значит, распределение величины  $F_1(X_1, W)$   $\varepsilon_2$ -близко к распределению  $F_1(X_1, U_{d_1})$ , которое, в свою очередь,  $\varepsilon_1$ -близко к равномерному, откуда распределение  $F_1(X_1, W)$   $(\varepsilon_1 + \varepsilon_2)$ -близко к равномерному, что и требовалось.  $\square$

### 4.3. Построение блочного источника

Описаны различные методы построения блочных источников. Один из первых появился в работе [10] и основан на попарно независимом выборе битов из исходного распределения. Другой метод описан в работе [8] и представлен в разделе 5. Этот метод был развит и усилен в работе [13].

## 5. Конструкция Та-Шмы

### 5.1. Мчрджеры

**Определение 5.1.** Случайная величина  $Z = Z_1 \cdots Z_b$  называется  $b$ -блочным где-то случайным  $(k, \varepsilon, \eta)$ -источником, если  $Z_i$  — случайные величины на  $\{0, 1\}^k$  и существует случайная величина  $Y$  на  $\{0, \dots, b\}$ , такая что:

- Для всех  $i \in \{1, \dots, b\}$   $\text{dist}((Z_i|Y = i), U_k) \leq \varepsilon$ ;
- $\Pr[Y = 0] \leq \eta$ .

$Y$  называется  $(k, \varepsilon, \eta)$ -селектором для  $Z$ .

Несложно доказать следующую лемму:

**Лемма 5.1.** 1. Любой где-то случайный  $(k, \varepsilon, \eta)$ -источник  $(\varepsilon + \eta)$ -близок к некоторому где-то случайному  $(k, 0, 0)$ -источнику.

2. Если  $Z$  — где-то случайный  $(k, 0, 0)$ -источник, то  $H_\infty(Z) \geq k$ .

Дадим определение мчрджера.

**Определение 5.2.** Функция  $M: (k)^b \times (d) \rightarrow (m)$  называется  $\varepsilon$ -мчрджером, если для любого  $b$ -блочного где-то случайного  $(k, 0, 0)$ -источника  $Z$  распределение  $M(Z, U_d)$   $\varepsilon$ -близко к равномерному.

### 5.2. Композиция двух экстракторов посредством мчрджера

**Определение 5.3.** Пусть  $E_1: (n) \times (d_1) \rightarrow (d_2)$  и  $E_2: (n) \times (d_2) \rightarrow (m_2)$  — экстракторы, а  $M: (m_2)^n \times (\mu_1) \rightarrow (m)$  — мчрджер. Тогда композицией экстракторов посредством мчрджера называется функция  $E_2 \overset{M}{\odot} E_1: (n) \times (d_1 + \mu_1) \rightarrow (m)$ , определяемая следующим образом. Пусть  $a \in \{0, 1\}^n$ ,  $r_1 \in \{0, 1\}^{d_1}$ ,  $r_2 \in \{0, 1\}^{\mu_1}$ . Положим для  $i = 1, \dots, n$   $q_i = E_1(a_{[i, n]}, r_1)$ , а  $z_i = E_2(a_{[1, i-1]}, q_i)$ . Обозначим  $E_2 \ominus E_1 = z_1 \cdots z_n$  и положим  $E_2 \overset{M}{\odot} E_1(a, r_1, r_2) = M(E_2 \ominus E_1, r_2)$ .

*Замечание.* Вообще говоря, слова  $a_{[i, n]}$  и  $a_{[1, i-1]}$  короче  $n$  битов. Однако, поскольку мы рассматриваем распределения на этих словах, и интересуемся только их минимальной энтропией, мы можем формально дополнить эти слова до нужной длины, например нулями.

*Замечание.* Можно считать, что  $m_1 \geq d_2$ , поскольку свойство близости к равномерному распределению сохраняется при взятии ограничения на часть битов.

**Теорема 5.2.** Пусть  $E_1$  —  $(k_1, \zeta_1)$ -экстрактор,  $E_2$  —  $(k_2, \zeta_2)$ -экстрактор, а  $M$  —  $\zeta_3$ -мчрджер. Тогда для любого параметра  $s > 0$   $E_2 \overset{M}{\odot} E_1$  —  $(k_1 + k_2 + s, \zeta_1 + \zeta_2 + \zeta_3 + 8n2^{-s/3})$ -экстрактор.

*Доказательство.* Очевидно, достаточно доказать, что  $E_2 \ominus E_1$  — где-то случайный  $(m_2, \zeta_1 + \zeta_2, 8n2^{-s/3})$ -источник. Пусть  $X$  — случайная величина с  $H_\infty(X) \geq k_1 + k_2 + s$ . Обозначим через  $Q_i$  и  $Z_i$  случайные величины, принимающие значения  $q_i$  и  $z_i$  соответственно. Положим  $\varepsilon_3 = 2^{-s/3}$ ,  $\varepsilon_2 = 2\varepsilon_3$ ,  $\varepsilon_1 = 2\varepsilon_2$ .

Определим селектор для величины  $Z = Z_1 \cdot Z_2 \cdot \dots \cdot Z_n = E_2 \ominus E_1$ . Пусть  $w \in \{0, 1\}^n$ , и

$$f(w) = \max\{i: \Pr [X_{[i,n]} = w_{[i,n]} | X_{[1,i-1]} = w_{[1,i-1]}] \leq (\varepsilon_2 - \varepsilon_3) \cdot 2^{-k_1}\}$$

Это уже почти селектор, но его надо немного подправить, избавившись от слишком редко принимаемых значений. Ведь если значение принимается редко, то соответствующее условное распределение может вести себя как угодно, а не быть близким к равномерному. Более строго: назовем  $w$  плохим, если  $f(w) = i$  и

1.  $\Pr [f(x) = i] \leq \varepsilon_1$ , или
2.  $\Pr [f(x) = i \mid x_{[1,i-1]} = w_{[1,i-1]}] \leq \varepsilon_2$ , или
3.  $\Pr [x_i = w_i \mid x_{[1,i-1]} = w_{[1,i-1]}] \leq \varepsilon_3$ .

Обозначим через  $B$  множество всех плохих  $w$ , а через  $B_i$  — множество всех  $w$ , удовлетворяющих условию  $i$ . Теперь определим селектор как

$$Y(w) = \begin{cases} 0, & \text{если } w \text{ плохое;} \\ f(w), & \text{иначе.} \end{cases}$$

Нетрудно доказать, что доля плохих  $w$  не превосходит  $n(\varepsilon_1 + \varepsilon_2 + \varepsilon_3) \leq 8n2^{-s/3}$ . Осталось доказать, что  $(Z_i | Y = i)$   $(\zeta_1 + \zeta_2)$ -близко к равномерному. Это следует из двух утверждений:

**Утверждение 5.3.** Если  $\Pr [Y = i \mid X_{[1,i-1]} = w_{[1,i-1]}] > 0$ , то  $H_\infty(X_{[i,n]} \mid Y = i \text{ и } X_{[1,i-1]} = w_{[1,i-1]}) \geq k_1$ .

**Утверждение 5.4.**  $H_\infty(X_{[1,i-1]} \mid Y = i) \geq k_2$ .

Действительно, для всех  $w_{[1,i-1]}$ , удовлетворяющих условию утверждения 5.3, распределение  $Q_i | Y = i$  и  $X_{[1,i-1]} = w_{[1,i-1]}$   $\zeta_1$ -близко к равномерному (по свойству экстрактора  $E_1$ ). Отсюда распределение  $(X_{[1,i-1]} | Y = i) \times (Q_i | Y = i \text{ и } X_{[1,i-1]} = w_{[1,i-1]})$   $\zeta_1$ -близко к  $(X_{[1,i-1]} | Y = i) \times U_{d_2}$ . Отсюда по свойству экстрактора  $E_2$  получаем, что  $(Z_i | Y = i)$   $(\zeta_1 + \zeta_2)$ -близко к равномерному.  $\square$

Докажем теперь утверждения 5.3 и 5.4

*Доказательство утверждения 5.3.* Для любого  $w$ , такого что  $Y(w) = i$ , выполнено

$$\begin{aligned} \Pr [x_{[i,n]} = w_{[i,n]} \mid x_{[1,i-1]} = w_{[1,i-1]}, Y(x) = i] &\leq \frac{\Pr [X_{[i,n]} = w_{[i,n]} \mid x_{[1,i-1]} = w_{[1,i-1]}}{\Pr [Y(x) = i \mid X_{[1,i-1]} = w_{[1,i-1]}} \leq \\ &\leq \frac{(\varepsilon_2 - \varepsilon_3) \cdot 2^{-k_1}}{\Pr [Y(x) = i \mid x_{[1,i-1]} = w_{[1,i-1]}} \leq \frac{(\varepsilon_2 - \varepsilon_3) \cdot 2^{-k_1}}{\varepsilon_2 - \varepsilon_3} = 2^{-k_1} \end{aligned}$$

Первое неравенство следует из того, что  $\Pr [A|B] \leq \Pr [A] / \Pr [B]$ , второе — из того, что  $f(w) = i$ , и определения  $f$ . Докажем третье, т.е. что  $\Pr [Y(x) = i \mid x_{[1,i-1]} = w_{[1,i-1]}] \geq \varepsilon_2 - \varepsilon_3$ , если только не равно нулю. Действительно, если  $w_{[1,i-1]}$  служит началом для некоторого  $w$  с  $Y(w) = i \neq 0$ , то никакое продолжение  $w_{[1,i-1]}$  не может быть плохим по первому и второму условию. Значит,  $\Pr [Y(x) = i \mid x_{[1,i-1]} = w_{[1,i-1]}] = \Pr [f(x) = i \mid x_{[1,i-1]} = w_{[1,i-1]}]$ .  $\Pr [f(x) = i \mid x \in B_3 \mid x_{[1,i-1]} = w_{[1,i-1]}] \geq \varepsilon_2 - \varepsilon_3$ . Первое слагаемое не меньше  $\varepsilon_2$ , поскольку  $x \notin B_2$ . Оценка на второе тоже понятна: для тех  $x$ , где  $f(x) = i$ , соответствующая вероятность не превосходит  $\varepsilon_3$  по определению  $B_3$ , а для остальных и вовсе равна нулю.  $\square$

*Доказательство утверждения 5.4.* Пусть  $w_{[1,i-1]}$  — произвольное слово, продолжаемое до  $w$  с  $Y(w) = i$ . Оценим вероятность  $\Pr [x_{[1,i-1]} = w_{[1,i-1]}]$ .

$$\begin{aligned} \Pr [x_{[1,i-1]} = w_{[1,i-1]}] &= \frac{\Pr [x_{[1,n]} = w_{[1,n]}]}{\Pr [x_{[i,n]} = w_{[i,n]} \mid x_{[1,i-1]} = w_{[1,i-1]}]} = \\ &= \frac{\Pr [x_{[1,n]} = w_{[1,n]}]}{\Pr [x_i = w_i \mid x_{[1,i-1]} = w_{[1,i-1]}] \cdot \Pr [x_{[i+1,n]} = w_{[i+1,n]} \mid x_{[1,i]} = w_{[1,i]}]} \end{aligned}$$

Числитель оценивается сверху как  $2^{-(k_1+k_2+s)}$ , поскольку  $H_\infty(X) \geq k_1 + k_2 + s$ . Первый сомножитель знаменателя оценивается снизу как  $\varepsilon_3$ , поскольку  $w \notin B_3$ . Наконец, второй сомножитель знаменателя оценивается снизу как  $(\varepsilon_2 - \varepsilon_3) \cdot 2^{-k_1}$ , поскольку  $f(w) = i$ . В итоге имеем

$$\Pr [x_{[1,i-1]} = w_{[1,i-1]}] \leq \frac{2^{-k_2-s}}{\varepsilon_3(\varepsilon_2 - \varepsilon_3)}. \quad (4)$$

Далее,

$$\begin{aligned} \Pr [x_{[1,i-1]} = w_{[1,i-1]} \mid Y(x) = i] &\leq \frac{\Pr [x_{[1,i-1]} = w_{[1,i-1]}]}{\Pr [Y(x) = i]} \leq \\ &\leq \frac{2^{-k_1-s}}{\varepsilon_3(\varepsilon_2 - \varepsilon_3) \Pr [Y(x) = i]} \leq \frac{2^{-k_1-s}}{\varepsilon_3(\varepsilon_2 - \varepsilon_3)(\varepsilon_1 - \varepsilon_2 - \varepsilon_3)} = 2^{-k_1} \end{aligned}$$

Последнее неравенство следует из того, что если  $\Pr [Y(x) = i] > 0$ , то  $\Pr [f(x) = i] \geq \varepsilon_1$ . Исключив  $x \in B_2$  и  $x \in B_3$ , получаем нужную оценку  $\Pr [Y(x) = i] \geq \varepsilon_1 - \varepsilon_2 - \varepsilon_3$ .  $\square$



### 5.3. Композиция нескольких экстракторов

Распространим нашу технику на произвольное число экстракторов.

**Определение 5.4.** Пусть  $E_i: (n) \times (d_i) \rightarrow (d_{i+1} + s_{i+1})$  суть  $(k_i, \zeta_i)$ -экстракторы для  $i = 1, \dots, t$ ,  $s_i \geq 0$ , а  $s_2 = 0$ . Пусть также  $M_i: (d_{i+2} + s_{i+2})^n \times (\mu) \rightarrow (d_{i+2})$  суть  $\bar{\zeta}_i$ -мчрджеры для  $i = 1, \dots, t-1$ . Определим функцию  $E = E_t \overset{M_{t-1}}{\odot} E_{t-1} \overset{M_{t-2}}{\odot} \dots \overset{M_1}{\odot} E_1: (n) \times (d_1 + \mu_1 + \dots + \mu_{t-1}) \rightarrow (d_{t+1})$  индуктивно по правой ассоциативности:  $E := E_t \overset{M_{t-1}}{\odot} \left( E_{t-1} \overset{M_{t-2}}{\odot} \dots \overset{M_1}{\odot} E_1 \right)$ .

**Теорема 5.5.** Для любого параметра безопасности  $s > 0$   $E$  является  $(\sum_{i=1}^t k_i + (t-1)s, \sum_{i=1}^t \zeta_i + \sum_{i=1}^{t-1} \bar{\zeta}_i + (t-1)n2^{-s/3+3})$ -экстрактором. Если  $E_i$  и  $M_i$  вычислимы за полиномиальное время, то и  $E$  тоже.

*Доказательство.* Параметры экстрактора очевидным образом получаются применением по индукции теоремы 5.2. Докажем сохранение полиномиальной вычислимости. Будем действовать при помощи динамического программирования по следующему алгоритму:

1. Вход:  $x \in \{0, 1\}^n$ ,  $y \in \{0, 1\}^{d_1}$  и  $y_j \in \{0, 1\}^{\mu_j}$ ,  $j = 1, \dots, t-1$ .
2. Будем вычислять матрицу  $M$  с элементами  $M_{ji} = \left( E_j \overset{M_{j-1}}{\odot} \dots \overset{M_1}{\odot} E_1 \right) (x_{[i, n]}, y y_1 \dots y_{j-1})$  для  $1 \leq i \leq n$  и  $1 \leq j \leq t$ .

Первый ряд матрицы,  $M_{1i}$ , может быть вычислен непосредственно как  $E_1(x_{[i, n]}, y)$ . Пусть мы заполнили  $j$ -ый ряд матрицы, заполним  $(j+1)$ -ый.

- Обозначим  $q_l = M_{jl}$ ,  $l = i, \dots, n$ , и положим  $z_l = E_{j+1}(x_{[i, l-1]}, q_l)$ .
- Положим  $M_{j+1, l} = M_j(z_i \dots z_n, y_j)$ .

Результат вычислений будет правильным по определению мчрджера, полиномиальность времени работы понятна.  $\square$

### 5.4. Построение мчрджеров

Опишем конструкцию явного построения мчрджеров. Вначале заметим, что любой  $(k, \varepsilon)$ -экстрактор с  $n = 2k$  «извлекает случайность» из любой случайной величины  $X$  с  $H_\infty(X) \geq k$ , в частности, из двухблочного где-то случайного  $(k, 0, 0)$ -источника. (по лемме 5.1) Таким образом, он является двухблочным мчрджером. Построим  $b$ -блочный мчрджер на базе двухблочных. А именно, будем действовать так:

**Алгоритм 5.1.** Пусть  $M: (k)^2 \times (d(k)) \rightarrow (k - m(k))$  — мчрджер. Построим рекурсивно мчрджер  $M_l: (k)^{2^l} \times (l \cdot d(k)) \rightarrow (k - l \cdot m(k))$ :

1. Вход:  $x^l = x_1^l \dots x_{2^l}^l$ , где  $x_i^l \in \{0, 1\}^k$ ;  $d = d_1 \dots d_l$ ,  $d_i \in \{0, 1\}^{d(k)}$ .
2. Если  $l = 0$ , возвращаем  $x^l$ .
3. Иначе положим  $x_i^{l-1} = M(x_{2i-1}^l, x_{2i}^l, d_l)$ ,  $i = 1, \dots, 2^{l-1}$ .
4. Возвратим  $M_{l-1}(x_1^{l-1} \dots x_{2^{l-1}}^{l-1}, d_1 \dots d_{l-1})$ .

Докажем корректность работы алгоритма.

**Теорема 5.6.** Пусть для всех  $k$  для некоторых монотонно растущих функций  $d$ ,  $m$  и  $\varepsilon^{-1}$  существует полиномиально вычислимый  $\varepsilon(k)$ -мчрджер  $M: (k)^2 \times (d(k)) \rightarrow (k - m(k))$ . Тогда  $M_l$ , построенный по алгоритму 5.1, является  $l \cdot \varepsilon(k - m(k))$ -мчрджером.

*Доказательство.* Для  $j = l, \dots, 0$  и  $i = 1, \dots, 2^j$  обозначим через  $X_i^j$  случайную величину, принимающую значения  $x_i^j$  с вероятностями, индуцированными распределением  $X$  на  $x = x^l \in \{0, 1\}^{k \cdot 2^l}$  и равномерным на  $d \in \{0, 1\}^{l \cdot d(k)}$ . Заметим, что  $X^l = X$  — исходное распределение, а  $X^0$  — выход. Через  $X^j$  обозначим конкатенацию  $X^j = X_1^j \dots X_{2^j}^j$ . Обозначим  $k_j = k - (l - j)m(k)$  и докажем более общий факт: если  $X$  — где-то случайный  $(k, 0, 0)$ -источник, то для всех  $1 \leq i \leq 2^j$

$$\text{dist}((X_i^j | Y \in [2^{l-j}(i-1) + 1, 2^{l-j}]), U_{k_j}) \leq (l - j)\varepsilon(k_j),$$

где  $Y$  —  $(k, 0, 0)$ -селектор для  $X$ . Проведем доказательство нисходящей индукцией по  $j$ . При  $j = l$  утверждается, что  $\forall i \text{ dist}((X_i | Y = i), U_k) = 0$ , что верно по определению  $Y$ . Пусть мы доказали утверждение для  $j$ , докажем для  $j-1$ . По предположению индукции выполнено:

- $\text{dist}((X_{2i-1}^j | Y \in [2^{l-j}(2i-2) + 1, 2^{l-j}]), U_{k_j}) \leq (l - j)\varepsilon(k_j)$ ;
- $\text{dist}((X_{2i}^j | Y \in [2^{l-j}(2i-1) + 1, 2^{l-j}]), U_{k_j}) \leq (l - j)\varepsilon(k_j)$ .

Воспользуемся следующей леммой:

**Лемма 5.7.** Пусть  $A$ ,  $B$  и  $Y$  суть случайные величины. Предположим, что  $\text{dist}((A | Y \in S_1), U_k) \leq \varepsilon$  и  $\text{dist}((B | Y \in S_2), U_k) \leq \varepsilon$  для некоторых непересекающихся множеств  $S_1$  и  $S_2$ . Тогда распределение  $(AB | Y \in S_1 \cup S_2)$   $\varepsilon$ -близко к некоторому распределению  $W$  с  $H_\infty(W) \geq k$ .

По лемме имеем, что  $(X_{2i-1}^j X_{2i}^j | Y \in [2^{l-j}(2i-2) + 1, 2^{l-j}])$   $(l - j)\varepsilon(k_j)$ -близко к некоторому  $W$  с  $H_\infty(W) \geq k_j$ . Поскольку  $X_i^{j-1} = M(X_{2i-1}^j, X_{2i}^j, d_j)$ ,  $(X_i^{j-1} | Y \in [2^{l-j}(2i-2) + 1, 2^{l-j}])$   $(l - j)\varepsilon(k_j)$ -близко к  $M(W, d_j)$ , т. е.  $((l - j)\varepsilon(k_j) + \varepsilon(k_j))$ -близко к равномерному, откуда с учетом того, что  $\varepsilon(k_j) \leq \varepsilon(k_{j-1})$ , вытекает утверждение теоремы.  $\square$

*Доказательство леммы 5.7.* Достаточно рассмотреть случай, когда  $\Pr[Y \in S_1 \cup S_2] = 1$ , т. к. при ограничении на это множество условные распределения не изменятся. А в этом случае функция  $Z = i : Y \in S_i$  будет по условию  $(k, \varepsilon, 0)$ -селектором для  $AB$ . По лемме 5.1  $AB$  будет  $\varepsilon$ -близко к некоторому распределению  $W$  с  $H_\infty(W) \geq k$ . В общем случае то же самое будет выполнено для  $(AB | Y \in S_1 \cup S_2)$ .  $\square$

## 5.5. Построение итогового экстрактора

**Лемма 5.8** ([15]). Для некоторой константы  $c > 1$  и любого  $k = \Omega(\log n)$  существует полиномиально вычислимый  $(2k, 2^{-k/5})$ -экстрактор  $A_k: (n) \times (k) \rightarrow (ck)$ . Обозначим эту константу  $c_{sz}$ .

Покажем, как можно построить хороший экстрактор, имея хороший мчрджер.

**Лемма 5.9.** Пусть для всех  $k \in [k', k'']$  существует полиномиально вычислимый  $\varepsilon$ -мчрджер  $M_k: (k)^n \times (d) \rightarrow (\alpha k)$ , где  $\alpha$  — константа в интервале  $(1/c_{sz}, 1)$ . Тогда для тех же  $k$  существует полиномиально вычислимый  $(k, \text{poly}(n) \cdot \varepsilon)$ -экстрактор  $E: (n) \times (O(k' \log(1/\varepsilon)) + d \log n) \rightarrow (\Omega(k))$

*Доказательство.* Положим  $b = c_{sz} \cdot \alpha$ ,  $k_i = b^i k' \log(1/\varepsilon)$ , а  $t$  — минимальное число, что  $\sum_{i=1}^t k_i \leq k/2$ . Определим  $E$  как  $E = E_t \overset{M_{t-1}}{\odot} E_{t-1} \dots \overset{M_1}{\odot} E_1$ , где

- $E_i: (n) \times (k_i) \rightarrow (c_{sz} k_i) - (2k_i, 2^{k_i/5})$ -экстрактор из леммы 5.8.
- $M_i: (c_{sz} k_{i+1})^n \times (d) \rightarrow (k_{i+2}) - \varepsilon$ -мчрджер, существующий по предположению леммы. ( $k_{i+2} = b k_{i+1} = \alpha \cdot c_{sz} k_{i+1}$ )

По теореме 5.5, примененной для  $d_i = k_i$  и  $s_i = (c_{sz} - b)k_{i-1}$ ,  $E$  является экстрактором. Проверим, что он имеет нужные параметры. Выберем параметр безопасности  $s = k/2t$ , тогда по выбору  $t$  минимальная энтропия  $E$  будет равна  $\sum_{i=1}^t k_i + (t-1)s > k$ . Очевидно,  $t = O(\log n)$ , это даст нужную оценку на количество случайных битов. Далее,  $k_{t+1} = b^{t+1} k' \log(1/\varepsilon) = \Omega\left(\frac{b^{t+1}-1}{b-1} k' \log(1/\varepsilon)\right) = \Omega(\sum_{i=1}^t k_i) = \Omega(k)$ . Наконец, ошибка  $E$  равна  $\sum_{i=1}^t 2^{-k_i/5} + (t-1)\varepsilon + (t-1)s2^{-s/3+3}$ , что с учетом того, что  $k = O(\log(1/\varepsilon))$ , даст требуемую оценку. Для полного доказательства осталось заметить, что полиномиальная вычислимость  $E$  также следует из теоремы 5.5.  $\square$

Сопоставляя результаты теоремы 5.6 и леммы 5.9, получаем, что для всех  $k$  существует полиномиальный  $(k, \varepsilon)$ -экстрактор  $E: (n) \times (d \text{polylog}(n) \log(1/\varepsilon)) \rightarrow (\Omega(k))$ , где  $d$  — количество случайных битов, необходимых для работы мчрджера. Однако, этот экстрактор «извлекает» еще не всю минимальную энтропию исходного распределения, а лишь какую-то фиксированную долю. Можно модифицировать алгоритм, чтобы «извлечь случайность» полностью. Для этого нам достаточно построить  $(k, \varepsilon)$ -экстрактор  $E: (2k) \times (d) \rightarrow (k - O(k/\log n))$ .

Покажем, как можно увеличить количество извлекаемых битов, используя один и тот же экстрактор. Пусть экстрактор  $E$  «извлекает случайность» из всех распределений с минимальной энтропией не меньше  $k$ . Что будет, если подать ему на вход распределение с минимальной энтропией  $K \geq k$ ? Можно действовать по следующему алгоритму: применять экстрактор много раз к одной и той же строке, каждый раз с новым набором случайных битов  $r_i$ , пока суммарная длина выхода не превысит  $K - k$ . В таком случае с высокой вероятностью условное распределение  $(X|E(x, r_1) \dots E(x, r_t))$  будет по-прежнему иметь минимальную энтропию не меньше  $k$ . Более точно утверждение можно сформулировать в виде двух лемм, доказанных в [8].

**Лемма 5.10.** *Предположим, что для некоторого  $k$  существует полиномиально вычислимый  $(k, \varepsilon)$ -экстрактор  $E_k: (n) \times (d) \rightarrow (t)$ . Тогда для всяких  $K \geq k$ , параметра безопасности  $s > 0$  и  $t \in N$  существует полиномиально вычислимый  $(K, t(\varepsilon + 2^{-s}))$ -экстрактор  $E: (n) \times (td) \rightarrow (\min\{tm, K - k - s\})$ .*

**Лемма 5.11.** *Предположим, что для всех  $k > k'$  существует полиномиально вычислимый  $(k, \varepsilon(n))$ -экстрактор  $E_k: (n) \times (d(n)) \rightarrow (k/f(n))$ . Тогда для любого  $k$  существует полиномиально вычислимый  $(k, f(n) \log n(\varepsilon + 2^{-d(n)}))$ -экстрактор  $E: (n) \times (O(f(n) \log n \cdot d(n))) \rightarrow (k - k')$ .*

Сопоставив результаты лемм 5.9 и 5.11, получаем следующее

**Следствие 5.12.** *Пусть для всякого  $k \geq k' = k'(n)$  существует полиномиально вычислимый  $\varepsilon$ -мчрджер  $M: (k)^n \times (d) \rightarrow (\alpha k)$ , где  $1/c_{sz} < \alpha < 1$ . Тогда для всякого  $k$  существует полиномиально вычислимый  $(k, \text{poly}(n)\varepsilon)$ -экстрактор  $E: (n) \times (O(k' \log n \log(1/\varepsilon) + \log^2 n \cdot d)) \rightarrow (k)$ .*

(недостающие  $k'$  битов можно скопировать непосредственно из случайных, от этого параметры не ухудшатся)

Теперь построим необходимые мчрджеры. В [15] построены следующие экстракторы:

**Лемма 5.13.** *Пусть  $k(n) \geq n^{1/2+\gamma}$  для некоторого  $\gamma > 0$ . Тогда для любого  $\varepsilon$  существует полиномиально вычислимый  $(k(n), \varepsilon)$ -экстрактор  $E: (n) \times (O(\log^2 n \log(1/\varepsilon))) \rightarrow (k^2(n)/n)$*

На их базе можно построить нужные мчрджеры. А именно, верна следующая

**Лемма 5.14.** *Пусть  $b > 1$ , а  $f(k) = f(k(n))$  — такая функция, что  $f(k) \leq \sqrt[3]{k}$  и для любого  $k \geq k_0(n)$   $f(k) \geq b \log n$ . Тогда для всех  $k \geq k_0$  существует  $\varepsilon$ -мчрджер  $M: (k)^n \times (\log n \text{polylog}(k) \cdot f^2(k) \log(1/\varepsilon)) \rightarrow (k - k/b)$ ,*

*Доказательство.* По лемме 5.13 существует полиномиально вычислимый  $(k/f(k), \varepsilon)$ -экстрактор  $E: (2k) \times (O(\log^2 k \log(1/\varepsilon))) \rightarrow k/f^2(k)$ . (Здесь используется неравенство  $f(k) \leq \sqrt[3]{k}$ )

Далее, по лемме 5.10 существует полиномиально вычислимый  $(k, \text{poly}(k) \cdot \varepsilon)$ -экстрактор  $E: (2k) \times (O(f^2(k) \log^2 k \log(1/\varepsilon))) \rightarrow (k - k/f(k))$ .

Наконец, по теореме 5.6 существует полиномиально вычислимый  $(\log n \text{poly}(k) \cdot \varepsilon)$ -мчрджер  $M: (k)^n \times (O(\log n \text{polylog}(k) \cdot f^2(k) \log(1/\varepsilon))) \rightarrow (k - \log n \cdot k/f(k))$ . Поскольку  $k/f(k) \leq k/b \log n$  для всех  $k \leq k_0$ , имеем  $\log n \cdot k/f(k) \leq k/b$ , что и требовалось.  $\square$

Эта лемма позволяет построить экстракторы, которые полностью «извлекают случайность» из распределений с высокой минимальной энтропией. Вначале заметим, что мы можем получить требуемые мчрджеры:

**Следствие 5.15.** *Для всех  $k \geq 2^{\sqrt{\log n}}$  существует  $(\text{polylog}(n) \cdot \varepsilon)$ -мчрджер*

$$M_k: (k) \times (\text{polylog}(n) \log(1/\varepsilon)) \rightarrow (\Omega(k)).$$

*Доказательство.* Возьмем  $f(k) = \log^c k$  для некоторого  $c > 2$ . Тогда при любом  $b$ ,  $k \geq 2^{\sqrt{\log n}}$  и достаточно большом  $n$  имеем  $\log^c k \geq b \log n$ , поэтому можно применить лемму 5.14.  $\square$

Подставив этот результат в следствие 5.12, получаем

**Следствие 5.16.** *Для любого  $k$  существует  $(k, \text{poly}(n) \cdot \varepsilon)$ -экстрактор*

$$B_k: (n) \times (O(2^{\sqrt{\log n}} \text{polylog}(n) \log(1/\varepsilon))) \rightarrow (k).$$

Полученный экстрактор «извлекает случайность» полностью, но использует сверхполилогарифмическое число случайных битов. Мы можем уменьшить это число методом композиции экстракторов. Правда, за счет этого вновь вырастет требуемая минимальная энтропия. А именно,

**Лемма 5.17.** *Пусть  $\varepsilon \geq 2^{-n^\gamma}$  для некоторого  $\gamma < 1$ . Тогда существует  $\beta < 1$ , такая что для всех  $k \geq n^\beta$  существует полиномиально вычислимый  $(k, \text{poly}(n) \cdot \varepsilon)$ -экстрактор  $E: (n) \times (\text{polylog}(n) \log(1/\varepsilon)) \rightarrow (\Omega(k))$*

*Доказательство.* Положим  $\delta = (1 - \gamma)/2$  и  $\beta = 1 - \delta/2$ . Возьмем  $E = B_k \overset{M}{\odot} E_{sz}$ , где

- $E_{sz}$  —  $(n^\beta, \varepsilon)$ -экстрактор  $E: (n) \times (O(\log^2 n \log(1/\varepsilon))) \rightarrow (n^{2\beta-1})$  из леммы 5.13;
- $B_k$  — экстрактор из следствия 5.16;
- $M$  — мчрджер из следствия 5.15.

Поскольку  $n^{2\beta-1} = n^\delta n^\gamma = \Omega(2^{\sqrt{\log n}} \log(1/\varepsilon))$ , экстрактор  $E$  корректно определен. По теореме 5.2  $E$  является полиномиально вычислимым  $(k + n^\beta + n^\gamma, \text{poly}(n) \cdot \varepsilon)$ -экстрактором  $E: (n) \times (\text{polylog}(n) \log(1/\varepsilon)) \rightarrow (\Omega(k))$ . В частности, если  $H_\infty(X) = \Omega(n^\beta)$ , мы извлечем  $\Omega(H_\infty(X))$  битов, что и требовалось в теореме.  $\square$

Наконец, мы можем вновь, сохранив полилогарифмическое число случайных битов, перейти к произвольной минимальной энтропии и получить итоговую теорему.

**Теорема 5.18.** *Для любых  $\gamma < 1$ ,  $\varepsilon \geq 2^{-n^\gamma}$  и  $k = k(n)$  существует полиномиально вычислимый  $(k, \varepsilon)$ -экстрактор  $E: (n) \times (\text{polylog}(n) \log(1/\varepsilon)) \rightarrow (k)$*

*Доказательство.* По лемме 5.11, лемма 5.17 влечет существование полиномиально вычислимого  $(n, \text{poly}(n) \cdot \varepsilon)$ -экстрактора  $E: (2n) \times (\text{polylog}(n) \log(1/\varepsilon)) \rightarrow (n - n^\beta)$ .

Далее, существует константа  $c$ , зависящая только от  $\gamma$ , такая что для всех  $\log^c n \leq k \leq n$  выполнено  $\log n \cdot k^\beta \leq k/\bar{c}$ , где  $\bar{c}$  таково, что  $1/c_{sz} < 1 - 1/\bar{c} < 1$ . Отсюда по теореме 5.6 для всякого  $k$  существует полиномиально вычислимый  $\text{poly}(n) \cdot \varepsilon$ -мчрджер  $M: (k)^n \times (\text{polylog}(n) \log(1/\varepsilon)) \rightarrow (k - k/\bar{c})$ .

Наконец, по следствию 5.12 получаем для любого  $k$  полиномиально вычислимый  $(k, \text{poly}(n) \cdot \varepsilon)$ -экстрактор  $E: (n) \times (\text{polylog}(n) \log(1/\varepsilon)) \rightarrow (k)$ . Полагая  $\varepsilon' = \varepsilon / \text{poly}(n)$ , получаем в точности утверждение теоремы.  $\square$

## 6. Экстрактор Тревисана

В 1999 году Л.Тревисан (Luca Trevisan) в статье [16] описал совершенно новую конструкцию экстракторов на основе понятий псевдослучайного генератора, комбинаторного дизайна и кодов, исправляющих ошибки. Эта конструкция гораздо проще всех предыдущих и (в усиленном виде, описанном в [12]) даёт лучшие параметры экстрактора.

### 6.1. Дизайны

Дизайном называется набор подмножеств некоторого множества, имеющих одинаковый размер и маленькие попарные пересечения. Более строго:

**Определение 6.1.** Пусть дано множество размера  $d$  (напомним, мы обозначали его  $[d]$ ). Семейство множеств  $S_1, \dots, S_m \subset [d]$ , каждое из которых имеет размер  $l$ , называется:

- $(l, \rho)$ -дизайном (design), если для всех  $i \neq j$  выполнено  $|S_i \cap S_j| \leq \log \rho$ ;
- слабым  $(l, \rho)$ -дизайном (weak design), если для всех  $j$  выполнено

$$\sum_{i < j} 2^{|S_i \cap S_j|} \leq \rho(m-1);$$

- равномерным слабым  $(l, \rho)$ -дизайном (uniform weak design), если для всех  $j$  выполнено

$$\sum_{i < j} 2^{|S_i \cap S_j|} \leq \rho(j-1).$$

Очевидно, любой дизайн является равномерным слабым дизайном с теми же параметрами, а равномерный слабый дизайн — слабым дизайном. Обратное, разумеется, неверно.

Известны следующие конструкции дизайнов:

**Лемма 6.1** ([16], [12]). Для всех  $l, m$  и  $\rho = \rho(l, m) > 1$  существует набор  $\mathcal{S}$  множеств  $S_1, \dots, S_m \subset [d]$ , каждое из которых имеет размер  $l$ , который является:

- $(l, \rho)$ -дизайном при  $d = O(l^2 m^{1/\rho})/\rho$ ;
- слабым  $(l, 1)$ -дизайном при  $d = O(l^2 \log m)$ ;
- равномерным слабым  $(l, \rho)$ -дизайном при  $d = O(l^2 / \log \rho)$ .

Более того, этот набор  $\mathcal{S}$  можно получить детерминированным алгоритмом за время  $O(2^d m)$  для дизайна и за время  $\text{poly}(m, d)$  для слабого и равномерного слабого дизайнов. Конструкции для слабого и равномерного слабого дизайнов обладают тем свойством, что любое их подмножество вида  $\{S_1, \dots, S_i\}$  также является соответственно слабым и равномерным слабым дизайном с теми же параметрами.

В работе [12] доказана также оптимальность последних двух оценок.

## 6.2. Псевдослучайный генератор Нисана-Вигдерсона

Следующая конструкция описана в работе [9]. Пусть даны множество  $S = \{s_1 < \dots < s_l\} \subset [d]$  и  $y \in \{0, 1\}^d$ . Пусть  $y_i$  —  $i$ -ый бит  $y$ . Обозначим через  $y|_S$  слово, образованное битами  $y$ , лежащими в  $S$ , т.е.  $y|_S := y_{s_1} \dots y_{s_l}$ .

**Определение 6.2.** Пусть даны функция  $f: \{0, 1\}^l \rightarrow \{0, 1\}$  и семейство  $\mathcal{S}$  подмножеств  $S_1, \dots, S_m \subset [d]$  одинакового размера  $l$ . Тогда генератором Нисана-Вигдерсона называется функция  $NW_{f, \mathcal{S}}: \{0, 1\}^d \rightarrow \{0, 1\}^m$ , определенная равенством

$$NW_{f, \mathcal{S}}(y) = f(y|_{S_1}) \dots f(y|_{S_m}).$$

В работе [9] показано, что если  $f$  является трудновычислимой функцией, а  $\mathcal{S}$  — дизайн, то функция  $NW_{f, \mathcal{S}}$  действительно является псевдослучайным генератором.

## 6.3. Коды, исправляющие ошибки

Обычно под кодами, исправляющими ошибки, понимают отображения из  $\{0, 1\}^n$  в  $\{0, 1\}^{\bar{n}}$ , такие что расстояние Хэмминга (т.е. количество различающихся битов) между любыми двумя кодовыми (т.е. лежащими в образе  $\{0, 1\}^n$ ) словами достаточно велико. Нам потребуется более слабое понятие кодов, допускающих «декодирование списком» (list-decoding), т.е. таких, что в окрестности каждого кодового слова лежит не более полиномиального числа кодовых слов. Мы будем пользоваться кодами, существование которых утверждает следующая

**Лемма 6.2.** Для всех  $n \in \mathbb{N}$  и  $\delta = \delta(n) > 0$  существует код  $EC: \{0, 1\}^n \rightarrow \{0, 1\}^{\bar{n}}$  для некоторого  $\bar{n} = \text{poly}(n, 1/\delta)$ , такой что любой (хэммингов) шар в  $\{0, 1\}^{\bar{n}}$  относительного радиуса  $1/2 - \delta$  содержит не более  $1/\delta^2$  кодовых слов. Более того, зная центр шара  $x$ , можно алгоритмически получить список прообразов этих слов (т.е. декодировать  $x$  списком) за время  $\text{poly}(n, 1/\delta)$ . Можно также считать, что  $\bar{n}$  является степенью двойки.

Классический пример такого кода — конкатенация кодов Адамара и Рида-Соломона, дающая  $\bar{n} = O(n^2/\delta^4)$ .

## 6.4. Конструкция Тревисана

Экстрактор Тревисана устроен следующим образом. Получив на вход слова  $x \in \{0, 1\}^n$  и  $y \in \{0, 1\}^d$ , мы сначала закодируем  $x$  кодом  $EC$  из леммы 6.2, получив слово длины  $2^l$ , которое будем понимать как таблицу значений некоторой булевой функции  $f$ . А затем применим к  $y$  генератор Нисана-Вигдерсона, построенный по  $f$  и некоторому (слабому) дизайну  $\mathcal{S}$ . Более формально,

**Определение 6.3.** Пусть  $n$ ,  $d$  и  $m$  суть натуральные числа, а  $\delta = \delta(n) > 0$ . Пусть  $EC_\delta: \{0, 1\}^n \rightarrow \{0, 1\}^{2^l}$  — код из леммы 6.2 для выбранного  $\delta$ , а  $\mathcal{S}$  — семейство подмножеств  $S_1, \dots, S_m \subset [d]$  одинакового размера  $l$ . Для  $x \in \{0, 1\}^n$  обозначим через  $\hat{x}$

функцию из  $\{0, 1\}^l$  в  $\{0, 1\}$ , заданную таблицей значений  $EC_\delta(x)$ . Тогда функцией Тревисана называется функция  $TR_{\delta, \mathcal{S}}: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ , заданная равенством

$$TR_{\delta, \mathcal{S}}(x, y) = NW_{\hat{x}, \mathcal{S}}(y) = \hat{x}(y|_{S_1}) \dots \hat{x}(y|_{S_m}). \quad (5)$$

При надлежащем выборе параметров функция Тревисана является экстрактором. А именно, верно следующее

**Утверждение 6.3** ([12]). Пусть  $k \leq n$ ,  $d$  и  $m$  суть натуральные числа, а  $\varepsilon = \varepsilon(n) > 0$ . Тогда функция Тревисана  $TR_{\delta, \mathcal{S}}$ , построенная для  $\delta = \varepsilon/4m$  и слабого  $(l, \rho)$ -дизайна  $\mathcal{S}$ , где  $\rho = (k - 3 \log(m/\varepsilon) - d - 3)/m$ , является  $(k, \varepsilon)$ -экстрактором. Более того, она является экстрактором в сильном смысле, т. е. к выданным битам можно приписать полученные случайные биты с сохранением свойства экстрактора.

Утверждение 6.3 позволяет получить следующую теорему:

**Теорема 6.4** ([12]). Пусть  $m \leq k \leq n$  суть натуральные числа, а  $\varepsilon > 0$ . Тогда существует полиномиально вычислимый  $(k, \varepsilon)$ -экстрактор  $Ext: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  для

- $d = O\left(\frac{\log^2(n/\varepsilon)}{\log(k/m)}\right)$ , или
- $d = O(\log^2(n/\varepsilon) \log(1/\gamma))$ , где  $1 + \gamma = k/(m - 1)$  и  $\gamma < 1/2$ .

Таким образом, чтобы «извлечь всю случайность», нам достаточно  $O(\log^2(n/\varepsilon) \log k)$  дополнительных случайных битов, а чтобы «извлечь некоторую постоянную долю случайности» —  $O(\log^2(n/\varepsilon))$  битов. Из свойств слабого дизайна следует, что экстрактор Тревисана будет префиксным.

## 7. Конструкция Рейнгольда-Шалтиэля-Вигдерсона

В 2001 году О. Рейнгольд (Omer Reingold), Р. Шалтиэль (Ronen Shaltiel) и А. Вигдерсон (Avi Wigderson) в статье [13] путём комбинирования всех предложенных ранее методов построили наилучшие из известных на сегодняшний день экстракторы. А именно, для всех  $k$  построены экстракторы с  $m = ck$  и  $d = O(\log n (\log \log n)^2)$  для произвольной константы  $c < 1$ . Лемма 5.11 влечёт существование для всех  $k$  экстракторов с  $m = k$  и  $d = O(\log^2 n (\log \log n)^2)$ . При построении этих экстракторов использовался такой инструмент, как конденсеры.

**Определение 7.1.** Функция  $Con: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^{n'}$  называется  $(k, k', \varepsilon)$ -конденсером, если для любого распределения  $X$  на  $\{0, 1\}^n$  с  $H_\infty(X) \geq k$  существует такое распределение  $Y$  на  $\{0, 1\}^{n'}$ , что  $H_\infty(Y) \geq k'$  и  $\text{dist}(Y, Con(X, U_d)) < \varepsilon$ .



Таким образом, в отличие от экстрактора, конденсер не «извлекает случайность», а лишь «конденсирует» её на некотором меньшем числе битов.

В работе [13] показано, как построить полиномиально вычислимые конденсеры, и как из них построить экстракторы путём последовательного применения. При построении конденсеров используются методы, похожие на анализ различных плохих множеств в теореме 5.2.

## 8. Теорема Мучника

### 8.1. Определение колмогоровской сложности

Дадим формальное определение колмогоровской сложности.

**Определение 8.1.** Пусть  $\varphi$  — некоторый алгоритм, который переводит пары двоичных слов в двоичные слова. Тогда условной колмогоровской сложностью  $KS_\varphi(A|B)$  слова  $A$  относительно слова  $B$  при способе описания  $\varphi$  называется длина кратчайшего слова  $P$ , такого что  $\varphi(P, B) = A$ .

Среди всех способов описания найдётся оптимальный. А именно, выполнено следующее

**Утверждение 8.1.** Существует такой способ описания  $\psi$ , что для любого способа описания  $\varphi$  и всех слов  $A$  и  $B$  выполнено  $KS_\psi(A|B) \leq KS_\varphi(A|B) + O(1)$ . При этом константа в  $O(1)$  зависит от  $\varphi$ , но не от  $A$  и  $B$ .

**Определение 8.2.** Условной колмогоровской сложностью  $KS(A|B)$  слова  $A$  относительно слова  $B$  называется  $KS_\psi(A|B)$ , где  $\psi$  — способ описания из предыдущего утверждения. Колмогоровской сложностью  $KS(A)$  слова  $A$  называется  $KS(A|\Lambda)$ , где  $\Lambda$  — пустое слово.

Изучают также колмогоровскую сложность с ограничением на ресурсы.

**Определение 8.3.** Пусть  $\varphi$  — некоторый алгоритм (для многоленточной машины Тьюринга), который переводит пары двоичных слов в двоичные слова. Тогда условной колмогоровской сложностью  $C_\varphi^{t,s}(A|B)$  слова  $A$  относительно слова  $B$  за время  $t$  на зоне  $s$  при способе описания  $\varphi$  называется длина кратчайшего слова  $P$ , такого что  $\varphi(P, B) = A$ , и  $\varphi(P, B)$  вычисляется за время  $t$  на зоне  $s$ .

Утверждение о существовании оптимального способа описания принимает следующий вид:

**Утверждение 8.2.** Существует такой способ описания  $\psi$ , что для любого способа описания  $\varphi$  существует такая константа  $c$ , что всех слов  $A$  и  $B$  выполнено  $C_\psi^{ct \log t, cs}(A|B) \leq C_\varphi^{t,s}(A|B) + c$ .

Любой способ описания, удовлетворяющий этому утверждению, мы будем называть оптимальным. В тех случаях, когда выбор конкретного оптимального способа описания важен, мы будем его уточнять. Обычно рассматривают сложность либо с ограничением только на время, либо с ограничением только на память. Условимся, что один индекс означает ограничение на время, а ограничение только на память будем записывать как  $C^{\infty, p}$ .

Для ограничения на ресурсы также рассматривают сложность различения, которая не даёт ничего нового для обычной сложности.

**Определение 8.4.** Пусть  $\varphi$  — некоторый алгоритм, который переводит тройки двоичных слов в  $\{0, 1\}$ . Тогда условной колмогоровской сложностью различения  $CD_{\varphi}^{t,s}(A|B)$  слова  $A$  относительно слова  $B$  за время  $t$  на зоне  $s$  при способе описания  $\varphi$  называется длина кратчайшего слова  $P$ , такого что  $\varphi(P, U, B) = 1$ , если  $U = A$ ,  $\varphi(P, U, B) = 0$  для  $U \neq A$ , и  $\varphi$  вычисляется за время  $t$  на зоне  $s$ .

Будем говорить, что  $P$  — это программа, принимающая  $U$ , если  $\varphi(P, U, B) = 1$ , и отвергающая в противном случае. Безусловная и не зависящая от способа описания сложности различения вводятся аналогично предыдущим. Также можно определить сложность различения, релятивизованную относительно некоторого оракула. В работе [2] доказана следующая оценка:

**Теорема 8.3.** *Найдётся такой полином  $p(n)$ , что для любого множества  $S$  и всех  $A \in S^{\infty} = S \cap \{0, 1\}^n$  выполнено*

$$CD^{p, S}(A) \leq 2 \log |S^{\infty}| + O(\log n).$$

*Более того, найдётся программа, удовлетворяющая этой оценке, спрашивающая оракул только относительно своего входа и отвергающая его, получив отрицательный ответ.*

Сложность с ограничением на ресурсы и сложность различения можно рассматривать для вычислительных моделей, отличных от детерминированных машин Тьюринга. Дадим соответствующие определения, будем считать, что  $\varphi$  — оптимальный способ описания.

**Определение 8.5.** Вероятностной сложностью  $CBP^{t,s}(A|B)$  называется длина кратчайшей программы  $P$ , такой что  $\Pr[\varphi(P, B, r) = A] > 2/3$ , и  $\varphi(P, B, r)$  работает время  $t$  на зоне  $s$  для всех  $r$ .

Пусть  $\varphi_n$  — оптимальный способ описания среди недетерминированных машин Тьюринга.

**Определение 8.6.** Недетерминированной сложностью  $CN^{t,s}(A|B)$  называется длина кратчайшей программы  $P$ , такой что  $\varphi_n(P, B)$  заканчивает работу хотя бы для одной ветви алгоритма, возвращает  $x$ , если заканчивает работу, и работает время  $t$  на зоне  $s$ .

**Определение 8.7.** Сложностью Артура-Мерлина  $СAM^{t,s}(A|B)$  называется длина кратчайшей программы  $P$ , такой что вероятность того, что  $\varphi_n(P, B, r)$  заканчивает работу хотя бы для одной ветви и на всех таких ветвях возвращает  $x$ , больше  $2/3$ , и  $\varphi_n(P, B, r)$  работает за время  $t$  на зоне  $s$  для всех  $r$ .

Алгоритмы, использующие недетерминизм и случайность, удобно рассматривать как игру двух игроков: Артура и Мерлина. Артур обладает полиномиальными вычислительными способностями, а Мерлин обладает способностями магическими и может угадывать сертификаты для языков из NP. При этом и Артур, и Мерлин могут использовать случайность из общего источника. Мерлин стремится вынудить Артура вычислить неверно. Если с большой вероятностью это у него не получится, то язык лежит в АМ.

## 8.2. Доказательство теоремы Мучника при помощи экстракторов

Для доказательства теоремы 1.1 при помощи экстракторов нам потребуется следующая лемма, доказанная в статье Л. Фортноу [2]. Будем рассматривать экстрактор как двудольный граф.

**Лемма 8.4.** Пусть существует экстрактор с параметрами  $n, k, d, m, \epsilon$ . Пусть  $S$  — подмножество его левой доли размера  $K = 2^k$ . Назовём «плохими» элементы правой доли, имеющие больше  $2DK/M$  соседей из  $S$ , и элементы  $S$ , все соседи которых плохие. Тогда плохих элементов  $S$  не больше чем  $2\epsilon K$ .

*Доказательство.* Вначале оценим размер множества  $Y$  плохих элементов правой доли. По определению экстрактора имеем  $\epsilon > \frac{|E(S, Y)|}{DK} - \frac{|Y|}{M}$ , где  $E(S, Y)$  — количество рёбер, ведущих из  $S$  в  $Y$ . По выбору  $Y$  имеем  $E(S, Y) \geq |Y| \cdot \frac{2DK}{M}$ , откуда подстановкой получаем  $|Y| < \epsilon M$ . Далее, пусть все соседи множества  $X$  попали в множество  $Y$ . Рассмотрим равномерное распределение на множестве  $S$  и применим к нему наш экстрактор. По свойству экстрактора мы должны получить распределение,  $\epsilon$ -близкое к равномерному. Индуцированная экстрактором мера множества  $Y$  не меньше  $\frac{|X|}{|S|}$ , поэтому

$$\frac{|X|}{K} - \frac{|Y|}{M} < \epsilon, \quad (6)$$

откуда с учётом  $|Y| < \epsilon M$  получаем  $|X| < 2\epsilon K$ , что и требовалось.  $\square$

Докажем теорему Мучника в следующей формулировке:

**Теорема 8.5.** Пусть  $A$  и  $B$  — произвольные слова. Пусть существует экстрактор с параметрами  $n = l(A)$  (т. е.  $n$  равно длине  $A$ ),  $k = KS(A|B)$ ,  $d = \Omega(\log n)$ ,  $m = KS(A|B)$ ,  $\epsilon = 1/n^3$ . Тогда найдётся слово  $X$  длины не более  $KS(A|B) + O(1)$ , для которого  $KS(X|A) \leq d + 2 \log n + O(1)$  и  $KS(A|B, X) \leq d + 2 \log n + O(1)$ .

Таким образом, использование оптимальных экстракторов позволяет получить теорему Мучника в исходной формулировке, а использование одной из известных конструкций — теорему, в которой поправки  $O(\log n)$  заменены на  $\text{polylog}(n)$ , но кодирование осуществляется полиномиальным алгоритмом.

*Доказательство теоремы 8.5.* Пусть  $E$  — экстрактор, существование которого утверждается в условии теоремы. Будем считать, что выбор  $E$  полностью определяется параметрами  $n$  и  $m$ : либо экстрактор строится по ним полиномиальным алгоритмом, либо появляется первым в каком-нибудь естественном порядке перебора. Будем рассматривать левую долю  $E$  как множество всех слов длины  $n$  (среди которых есть  $A$ ), а правую — как множество всех слов длины  $m$ , среди которых мы будем искать  $X$ . Рассмотрим в левой доле множество  $S_B$  всех слов  $P$ , для которых  $KS(P|B) \leq m$ . Оно имеет размер  $O(2^m)$ , поэтому по лемме 8.4 доля «плохих»  $P$  (т.е. все соседи которых имеют больше  $2D$  соседей из  $S_B$ ) в нчм не превышает  $c\varepsilon = c/n^3$  для некоторой константы  $c$ . Покажем, что  $A$  не может быть «плохим». Действительно, свойство быть «плохим» перечислимо: мы можем перечислять множество  $S_B$  и непосредственной проверкой устанавливать, что слово плохое для уже перечисленной части  $S_B$ . Таким образом, если бы  $A$  было плохим, то при известном  $B$  его можно было бы задать числами  $m$ ,  $n$  и номером в переборе всех плохих слов, что дало бы  $KS(A|B) \leq 2\log n + (m - 3\log n + O(1)) < m$ , однако  $KS(A|B) = m$ , откуда имеем противоречие.

Значит, у  $A$  есть хороший сосед справа. Обозначим его через  $X$  и покажем, что он удовлетворяет требованиям. Действительно,  $KS(X|A) \leq 2\log n + d + O(1)$ :  $2\log n$  битов нужно для задания  $n$  и  $m$ ,  $d$  — для задания номера  $X$  среди соседей  $A$ . Аналогично при известных  $B$  и  $X$  для задания  $A$  достаточно указать  $n$ ,  $m$  и номер  $A$  среди соседей  $X$  внутри  $S$ : мы можем перечислять множество  $S$ , а значит, и множество соседей  $X$  из  $S$ . Существование требуемого в теореме  $X$  установлено, тем самым теорема доказана.  $\square$

### 8.3. Теорема Мучника в случае нескольких условий

Исходная теорема Мучника обобщается следующим образом:

**Теорема 8.6.** Пусть  $A$ ,  $B$  и  $C$  — произвольные слова сложности не более  $n$ , а  $k \geq l$  — натуральные числа, такие что  $KS(A|B) \leq k$  и  $KS(A|C) \leq l$ . Тогда найдётся такое слово  $X$  длины не более  $k + O(\log n)$ , что  $KS(X|A) \leq O(\log n)$ ,  $KS(A|B, X) \leq O(\log n)$  и  $KS(A|C, X|_l) \leq O(\log n)$  для префикса  $X|_l$  слова  $X$  длины  $l$ .

Использование экстракторов позволяет доказать следующую теорему:

**Теорема 8.7.** Пусть  $A$ ,  $B$  и  $C$  — произвольные слова сложности не более  $n$ , а  $k \geq l$  — натуральные числа, такие что  $KS(A|B) \leq k$  и  $KS(A|C) \leq l$ . Пусть существует префиксный экстрактор с параметрами  $n$ ,  $k$ ,  $d$ ,  $m = k$ ,  $\varepsilon = 1/n^3$ . Тогда найдётся такое слово  $X$  длины не более  $k + O(\log n)$ , что  $KS(X|A) \leq d + O(\log n)$ ,  $KS(A|B, X) \leq d + O(\log n)$  и  $KS(A|C, X|_l) \leq d + O(\log n)$ .

*Доказательство.* Вновь будем следовать доказательству исходной теоремы, переводя его на язык экстракторов. Как и прежде, можно считать, что длина слова  $A$  равна его сложности, то есть  $n$ .

Вновь будем рассматривать левую долю экстрактора  $E$ , существование которого утверждается в условии, как множество всех слов длины  $n$ , а правую — как множество всех слов длины  $m$ , среди которых мы будем искать  $X$ . Нам потребуется усиленный вариант леммы 8.4:

**Лемма 8.8.** *Пусть существует экстрактор с параметрами  $n, k, d, m, \epsilon$ . Пусть  $S$  — подмножество его левой доли размера  $K = 2^k$ . Назовём «плохими» элементы правой доли, имеющие больше  $2DK/M$  соседей из  $S$ , и элементы  $S$ , по крайней мере половина соседей которых плохие. Тогда плохих элементов  $S$  не больше чем  $4\epsilon K$ .*

*Доказательство.* Доказательство повторяет доказательство леммы 8.4 вплоть до неравенства (6), которое нужно заменить на  $\frac{|X|}{2K} - \frac{|Y|}{M} < \epsilon$ , что с учётом  $|Y| < \epsilon M$  повлечёт требуемую оценку  $|X| < 4\epsilon K$ .  $\square$

Вновь рассмотрим множества  $S_B$  и  $S_C$  слов  $P$ , имеющих условную сложность не больше  $k$  и  $l$  относительно  $B$  и  $C$  соответственно. По лемме 8.8 доля плохих слов в каждом из них не превышает  $c/n^3$ , поэтому  $A$  не может быть плохим ни для одного из них. Значит, для каждого из слов  $B$  и  $C$  у  $A$  больше половины хороших соседей. Следовательно, хотя бы один сосед будет хорошим для обоих. Его мы и возьмём в качестве  $X$ . Свойства  $KS(X|A) \leq d + O(\log n)$  и  $KS(A|B, X) \leq d + O(\log n)$  доказываются в точности как раньше, последнее свойство следует из определения префиксного экстрактора.  $\square$

Эту теорему можно обобщить на произвольное количество условий:

**Теорема 8.9.** *Пусть  $A, B_1, \dots, B_p$  — произвольные слова сложности не более  $n$ , а  $k_1 \geq \dots \geq k_p$  — натуральные числа, такие что  $KS(A|B_i) \leq k_i$  для  $i = 1, \dots, p$ . Пусть существует префиксный экстрактор с параметрами  $n, k, d, m = k, \epsilon = 1/pn^3$ . Тогда найдётся такое слово  $X$  длины не более  $k_1 + O(\log n)$ , что  $KS(X|A) \leq d + O(\log n)$ ,  $KS(A|B_i, X|_{k_i}) \leq d + O(\log n)$  для  $i = 1, \dots, p$ .*

*Замечание.* Поскольку для оптимального экстрактора  $d = O(\log(n/\epsilon))$ , то мы получим точность  $O(\log n)$  вместо  $d + O(\log n)$  лишь для полиномиальных  $p$ , как и в исходной теореме из [6].

## 8.4. Теорема Мучника для сложности с ограничением на память

Немного изменённая конструкция из теоремы 8.5 позволяет распространить утверждение теоремы Мучника на колмогоровскую сложность с полиномиальным ограничением на память.

**Теорема 8.10.** Пусть  $A$  и  $B$  — произвольные слова длины не более  $n$ , а  $p$  — произвольное число. Пусть для всех  $k \leq C^{\infty,p}(A|B)$  существует экстрактор  $E_k$ , вычисляемый на зоне  $\text{poly}(n)$ , с параметрами  $n, k, d, m = k, \varepsilon = 1/n^3$ . Тогда найдется слово  $X$  длины не более  $C^{\infty,p}(A|B)$ , такое что  $C^{\infty, \text{poly}(n)}(X|A) \leq d + O(\log n)$  и  $C^{\infty, 2p + \text{poly}(\log p, n)}(A|B, X) \leq d + O(\log n)$ .

Попытка напрямую распространить доказательство теоремы Мучника на сложность с ограничением на память наталкивается на трудность: для нахождения плохих слов для множества  $S_B = \{P \mid C^{\infty,p}(P|B) \leq C^{\infty,p}(A|B)\}$  требуется зона, большая  $p$ . А тогда мы не сможем доказать, что  $A$  хорошее: утверждения  $C^{\infty,p}(A|B) = k$  и  $C^{\infty,p'}(A|B) < k$  непротиворечивы при  $p' > p$ . Однако, эту трудность можно обойти: для хороших  $A$  построим хеш-значение старым способом, а для плохих рассмотрим новый экстрактор с меньшей правой долей, и в нчм сделаем (рекурсивно) то же самое. Таким образом получится итеративный процесс, на последнем шаге которого все слова будут хорошими. Опишем эту конструкцию формально.

**Определение 8.8.** Будем говорить, что множество  $S$  перечислимо на зоне  $p$ , если существует алгоритм, который по любому числу  $z \leq |S|$  находит  $z$ -ый элемент  $S$ , при таком входе заканчивает работу на зоне  $p$ , а при ином входе работает на большей зоне или останавливается без возвращения результата. Очевидно, это эквивалентно следующему: некоторый алгоритм, работая на зоне  $p$ , печатает все слова из  $S$  на выходной ленте и останавливается, или выходит за пределы зоны  $p$ , при этом занятая на выходной ленте зона не считается.

**Лемма 8.11.** Пусть множество  $S$ , содержащееся в левой доле экстрактора  $E_k$ , перечислимо на зоне  $p$ . Тогда множество  $\text{Bad}(S)$  плохих для  $S$  слов перечислимо на зоне  $p + 2 \log p + \text{poly}(n)$ .

*Доказательство.* Вначале покажем, как можно перечислить  $\text{Bad}(S)$  на зоне  $p + \log p + \text{poly}(n)$ , если  $p$  известно. Запуская алгоритм, перечисляющий  $S$ , и ограничивая зону его работы числом  $p$ , мы будем получать слова  $P \in S$ . Получив очередное слово, проверим, является ли оно плохим, и если является, то напечатаем на выходной ленте. Покажем, как осуществить такую проверку. Будем последовательно вычислять всех соседей слова  $P$  и проверять, являются ли они плохими. Если все являются, значит  $P$  плохое, иначе хорошее. Покажем, как проверить, является ли слово  $X$  из правой части плохим. Снова будем запускать алгоритм, перечисляющий  $S$ , ограничивая зону его работы числом  $p$ . У каждого вновь полученного слова будем вычислять всех соседей и считать, сколько из них совпадают с  $X$ . Если общее число совпадений превысило  $2D$ , значит  $X$  плохое. Иначе, т.е. если алгоритм, перечисляющий  $S$ , остановился или попытался выйти за пределы зоны, а число совпадений не превысило  $2D$ ,  $X$  хорошее.

Посчитаем использованную зону. Заметим, что в процессе вычисления, занимающего зону  $p$ , нам не нужно обращаться к другому вычислению, занимающему такую же зону. Значит, все такие вычисления мы можем проводить на одной и той же зоне  $p + \log p$ , где  $\log p$  нужно для контроля невыхода за зону  $p$ . Для вычисления соседей слова из

левой доли достаточно зоны  $\text{poly}(n)$ , для остальных вычислений (переборов и хранения промежуточных результатов) достаточно зоны  $O(n)$ . В общей сложности получаем зону  $p + \log p + \text{poly}(n)$ .

Теперь покажем, как сделать то же самое, не зная заранее  $p$ , на зоне  $p + 2 \log p + \text{poly}(n)$ . Обозначим через  $S_q$  множество, которое перечисляет алгоритм, перечисляющий  $S$ , с ограничением  $q$  на зону его работы. Очевидно, для  $q < q'$  верно  $S_q \subset S_{q'}$  и  $\text{Bad}(S_q) \subset \text{Bad}(S_{q'})$ . Воспользуемся этим фактом: будем последовательно запускать предыдущий алгоритм для  $q = 1, 2, 3, \dots$ . Если на этапе с ограничением  $q$  получено, что слово  $P$  плохое, проверим, являлось ли оно плохим на предыдущем этапе (с ограничением  $q-1$ ), и если не являлось, то напечатаем его на выходной ленте. Таким образом, плохие для  $S$  слова будут печататься в другом порядке, но каждое по одному разу. Рано или поздно алгоритм дойдѣт до  $q = p$ , и все плохие для  $S$  слова к этому времени будут перечислены. По сравнению с предыдущим алгоритмом, потребуется дополнительная зона  $\log p$  для организации перебора всех  $q$ , и  $O(n)$  для хранения промежуточных результатов во время проверки, являлось ли полученное слово плохим на предыдущем этапе. Таким образом, общая зона составит  $p + 2 \log p + \text{poly}(n)$ , что и требовалось.  $\square$

**Лемма 8.12.** Множество  $S_B = \{P \mid C^{\infty,p}(P|B) \leq k\}$ , лежащее в левой доле экстрактора  $E_k$ , перечислимо на зоне  $2p + 2 \log p + O(n)$ .

*Доказательство.* Как и в предыдущей лемме, предположим вначале, что перечисляющему алгоритму известно  $p$ . Будем перебирать все описания длины  $k$  и запускать на них оптимальный способ описания  $\psi$  с ограничением на зону  $p$ . Одновременно будем считать количество шагов алгоритма  $\psi$  и принудительно останавливать его работу, если это количество превысит  $2^p$  (это значит, что  $\psi$  заиклился). Если  $\psi$  заканчивает работу, то результат его работы лежит в  $S_B$ , и мы можем включить его в перечисление. Описанный алгоритм требует зоны  $p + \log p$  для моделирования работы  $\psi$ , зоны  $p$  для контроля заикливания и зоны  $O(n)$  для перебора и хранения промежуточных результатов, всего  $2p + \log p + O(n)$ .

При неизвестном заранее  $p$  применим тот же приѣм, что и в предыдущей лемме: будем запускать описанный алгоритм для ограничений на зону  $q = 1, 2, 3 \dots$  и печатать слова из  $S_B$ , как только они получены впервые.  $\square$

**Следствие 8.13.** Положим  $S_B^0 = S_B$  и  $S_B^i = \text{Bad}(S_B^{i-1})$  при  $i \leq 1$ . Тогда при всех  $i$  множество  $S_B^i$  перечислимо на зоне  $2p + \text{poly}(\log p, n)$ .

*Доказательство.* Поскольку на каждом этапе  $S_B^i$  уменьшается хотя бы в  $n^3/2$  раз, то при  $i > k/2 \log n$  все  $S_B^i$  пусты и, следовательно, перечислимы. Для меньших  $i$  утверждение выполнено по индукции в силу предыдущих двух лемм.  $\square$

Теперь докажем саму теорему.

*Доказательство теоремы 8.10.* Пусть  $A$  хорошее, т.е.  $A \in S_B \setminus \text{Bad}(S_B)$ . Тогда у  $A$  есть хороший сосед справа, который мы и возьмѣм в качестве  $X$ . Если  $A$  плохое, то построим новый экстрактор  $E_k$  для  $k = C^{\infty,p}(A|B) - 2 \log n$ . Если  $A$  хорошее в новом

экстракторе для множества  $S_B^1 = \text{Bad}(S_B)$ , то у него есть хороший сосед, возьмем его в качестве  $X$ , иначе применим ту же процедуру еще раз, и так далее, пока не придём к случаю  $k < 2 \log n$ , когда плохих не останется.

Проверим, что условия теоремы выполнены. Для получения  $X$  при известном  $A$  достаточно знать номер этапа, на котором  $A$  будет хорошим,  $n$ ,  $k$ , и номер  $X$  среди соседей  $A$  в экстракторе на этом этапе, всего  $d + O(\log n)$  битов. Поскольку экстрактор вычислим на полиномиальной зоне,  $X$  также получается на полиномиальной зоне. Для получения  $A$  при известных  $B$  и  $X$  достаточно знать номер этапа  $i$ ,  $n$ ,  $k$ , и номер  $A$  среди соседей  $X$  в множестве  $S_B^i$ , всего  $d + O(\log n)$  битов. Поскольку  $S_B^i$  перечислимо на зоне  $2p + \text{poly}(\log p, n)$ , то и множество лежащих в нём соседей  $X$  перечислимо, причём на такой же зоне: увеличение составит  $\text{poly}(n)$  за счёт необходимости вычислять экстрактор и хранить промежуточные результаты. Таким образом,  $C^{\infty, \text{poly}(n)}(X|A) \leq d + O(\log n)$  и  $C^{\infty, 2p + \text{poly}(\log p, n)}(A|B, X) \leq d + O(\log n)$ , что и требовалось.  $\square$

## 8.5. Теорема Мучника для сложности с ограничением на время

Попытка распространить доказательство теоремы Мучника на сложность с полиномиальным ограничением на время наталкивается на серьёзные трудности: совершенно непонятно, как можно находить плохие слова за полиномиальное время. Однако, другая техника, описанная в статье [3], позволяет доказать такую теорему:

**Теорема 8.14.** *Для всякого полинома  $p$  найдётся полином  $q$ , такой что для произвольных слов  $A$  и  $B$  длины не более  $n$ , таких что  $C^p(A|B) \leq k$ , найдётся слово  $X$  длины не более  $k + O(\log^3 n)$ , такое что  $C^q(X|A) \leq O(\log^3 n)$  и  $CAM^q(A|B, X) \leq O(\log n)$ .*

*Доказательство.* Доказательство опирается не на произвольный экстрактор, а на конструкцию Тревисана и в целом следует доказательству теоремы 3 работы [3].

По лемме 6.1 существует слабый  $(l, 1)$ -дизайн для  $d = O(l^2 \log m) = O(\log^3 n)$ . Рассмотрим функцию Тревисана  $TR_{\delta, 1}: \{0, 1\}^m \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  для такого дизайна,  $m = k + d + 1^3$  и  $\delta = 1/8m$ . Очевидно, образ множества  $S_B = \{u \in \{0, 1\}^n \mid C^p(u|B) \leq k\}$  занимает не больше половины  $\{0, 1\}^m$ . Обозначим через  $\mathcal{B}$  предикат «быть в образе  $S_B$ ». Очевидно, для любого  $u \in S_B$  имеем  $\Pr[\mathcal{B}(TR_{\delta, 1}(u, U_d)) = 1] - \Pr[\mathcal{B}(U_m) = 1] \geq 1/2$ , поскольку первая вероятность равна 1, а вторая не больше  $1/2$ . Распишем это более подробно, обозначив через  $\hat{u}$  образ  $u$  под действием кода, исправляющего ошибки:

$$\Pr_y[\mathcal{B}(\hat{u}(y|_{S_1})\hat{u}(y|_{S_2}) \dots \hat{u}(y|_{S_m})) = 1] - \Pr_{r_1, \dots, r_m}[\mathcal{B}(r_1 r_2 \dots r_m) = 1] \geq 1/2,$$

где  $S_1, \dots, S_m$  — используемый дизайн. Очевидно, для некоторого  $i$  будет выполнено

$$\begin{aligned} & \Pr_{y, r_{i+1}, \dots, r_m}[\mathcal{B}(\hat{u}(y|_{S_1}) \dots \hat{u}(y|_{S_{i-1}})\hat{u}(y|_{S_i})r_{i+1} \dots r_m) = 1] - \\ & - \Pr_{y, b, r_{i+1}, \dots, r_m}[\mathcal{B}(\hat{u}(y|_{S_1}) \dots \hat{u}(y|_{S_{i-1}})br_{i+1} \dots r_m) = 1] \geq 1/2m. \end{aligned}$$

---

<sup>3</sup>Фактически это уравнение на  $m$ , имеющее решение  $m = k + O(\log^3 n)$ .



Более того, мы можем фиксировать некоторым образом биты  $y$  вне  $S_i$ , чтобы сохранить это соотношение. Обозначим  $y|_{S_i}$  через  $x$ , тогда все  $\hat{u}(y|_{S_j})$  зависят только от  $|S_j \cap S_i|$  общих с  $x$  битов. Таким образом,  $\hat{u}(y|_{S_j})$  есть некоторая функция  $\hat{u}_j(x)$ , определяемая  $2^{|S_j \cap S_i|}$  битами. Все функции  $\hat{u}_1(x), \dots, \hat{u}_{i-1}(x)$  можно задать  $\sum_{j < i} 2^{|S_j \cap S_i|}$  битами, что по определению слабого дизайна не превосходит  $m$ . Эти биты, вычисленные для  $u = A$ , мы и возьмем в качестве  $X$ . Очевидно, при известном  $A$  нам достаточно знать  $m$ ,  $y$  (вне  $S_i$ ) и  $i$ , чтобы их вычислить. Все вычисления будут полиномиальными, поэтому  $C^q(X|A) \leq O(\log^3 n)$ .

Осталось показать, что при известных  $B$  и  $X$  нам достаточно небольшой дополнительной информации, чтобы полиномиально вычислить  $A$  протоколом АМ. Как мы уже установили, имеет место неравенство

$$\Pr_{x \in \{0,1\}^l, b, r \in \{0,1\}^{m-i}} [\mathcal{B}(\hat{u}_1(x) \dots \hat{u}_{i-1}(x) \hat{u}(x) r) = 1] - \Pr_{x \in \{0,1\}^l, b, r \in \{0,1\}^{m-i}} [\mathcal{B}(\hat{u}_1(x) \dots \hat{u}_{i-1}(x) b r) = 1] \geq 1/2m. \quad (7)$$

Обозначим  $\hat{u}_1(x) \dots \hat{u}_{i-1}(x) b r$  через  $F(x, b, r)$ . Положим  $g_b(x, r)$  равным  $b$ , если  $\mathcal{B}(F(x, b, r)) = 1$ , и равным  $1-b$  в противном случае. Покажем, что  $\Pr_{x, b, r} [\hat{u}(x) = g_b(x, r)] \geq 1/2 + 1/2m$ :

$$\begin{aligned} \Pr_{x, b, r} [\hat{u}(x) = g_b(x, r)] &= \Pr_{x, b, r} [g_b(x, r) = \hat{u}(x) | b = \hat{u}(x)] \Pr_{x, b, r} [b = \hat{u}(x)] + \\ &\quad + \Pr_{x, b, r} [g_b(x, r) = \hat{u}(x) | b \neq \hat{u}(x)] \Pr_{x, b, r} [b \neq \hat{u}(x)] = \\ &= \frac{1}{2} \Pr_{x, b, r} [\mathcal{B}(F(x, b, r)) = 1 | b = \hat{u}(x)] + \frac{1}{2} \Pr_{x, b, r} [\mathcal{B}(F(x, b, r)) = 0 | b \neq \hat{u}(x)] = \\ &= \frac{1}{2} + \frac{1}{2} (\Pr_{x, b, r} [\mathcal{B}(F(x, b, r)) = 1 | b = \hat{u}(x)] - \Pr_{x, b, r} [\mathcal{B}(F(x, b, r)) = 1 | b \neq \hat{u}(x)]) = \\ &= \frac{1}{2} + \frac{1}{2} (\Pr_{x, r} [\mathcal{B}(F(x, \hat{u}(x), r)) = 1] - \Pr_{x, r} [\mathcal{B}(F(x, 1 - \hat{u}(x), r)) = 1]) = \\ &= \frac{1}{2} + \Pr_{x, b, r} [\mathcal{B}(F(x, \hat{u}(x), r)) = 1] - \Pr_{x, b, r} [\mathcal{B}(F(x, b, r)) = 1] \geq \frac{1}{2} + \frac{1}{2m}. \end{aligned}$$

Можно положить  $b$  равным некоторому  $b_1 \in \{0, 1\}$  таким образом, чтобы сохранить это соотношение. Бит  $b_1$  можно включить в описание  $u$  при известных  $B$  и  $X$ . Без ограничения общности можно считать, что  $b_1 = 1$ , поэтому мы опустим этот индекс в дальнейших рассуждениях. Покажем, как можно (приблизительно) вычислить  $g(x, r)$  протоколом Артура-Мерлина.

Если бы мы знали, как можно фиксировать биты  $r$ , чтобы сохранить соотношение  $\Pr_x [\hat{u}(x) = g(x, r)] \geq \frac{1}{2} + \frac{1}{2m}$ , то мы могли бы найти  $\hat{u}$  недетерминированным протоколом без всякой случайности. Действительно, мы можем вычислить  $g(x, r)$  недетерминированным протоколом и получить для всех  $x$  строку  $\hat{v}$ , совпадающую с  $\hat{u}$  не меньше чем на доле  $\frac{1}{2} + \frac{1}{2m}$  от всех битов. Недетерминированность нужна, чтобы вычислять  $\mathcal{B}$ : в качестве сертификата того, что  $\mathcal{B}(Y) = 1$ , можно предъявить описание прообраза, лежащего в  $S_B$ , и номер ребра, ведущего из этого прообраза в  $Y$ . Поскольку запросы к  $\mathcal{B}$  неадаптивные, мы можем в самом начале указать число  $a$  положительных ответов на запросы, тогда нахождение  $a$  положительных ответов гарантирует, что все остальные

отрицательные. К сожалению, мы не знаем, как фиксировать  $r$ , и потому будем выбирать их случайными. Покажем, что в этом случае можно воспользоваться аналогичным рассуждением.

Скажем, что  $r$  даёт  $\alpha$ -приближение к  $\hat{u}$ , если  $\Pr_x[g(x, r) = \hat{u}(x)] \geq \alpha$ . Будем отождествлять  $g(x, r)$  со строкой  $z_r$ , в которой бит номер  $x$  равен  $b_1 = 1$  тогда и только тогда, когда  $g(x, r) = 1$ . Количество единиц в строке  $z_r$  (т.е. её вес,  $w(z_r)$ ) совпадает с количеством слов  $x$ , для которых  $\mathcal{B}(\hat{u}_1(x) \dots \hat{u}_{i-1}(x)1r) = 1$ .

Начнём описывать протокол: вначале Артур выбирает случайные строки  $r_1, \dots, r_s$  длины  $m - i$  для некоторого полинома  $s = s(n)$ . Включим в описание среднее число  $\bar{a} = 2^{m-i} \sum_{x,r} g(x, r)$  положительных значений  $\mathcal{B}$  среди всех  $r$ . Будем требовать, чтобы реальное число положительных значений  $\mathcal{B}$  было близко к его ожиданию  $s\bar{a}$ . Покажем, что вероятность этого велика.

**Утверждение 8.15.** *Для любого  $\gamma = \gamma(\bar{n}, m) > 0$  найдётся  $s = O(\bar{n}^2/\gamma^2)$ , такой что с вероятностью не меньше  $3/4$  (по выбору  $r_1, \dots, r_s$ ) выполнены два условия:*

1. Доля  $1/8m$  от  $r_1, \dots, r_s$  даёт  $(\frac{1}{2} + \frac{1}{4m})$ -приближение к  $\hat{u}$ .
2. Общее число положительных значений  $\mathcal{B}$  на строках  $r_1, \dots, r_s$  лежит в пределах  $\gamma s$  от ожидания:

$$\left| \sum_{j=1}^s w(z_j) - s\bar{a} \right| \leq \gamma s.$$

*Доказательство.* Оценим сверху вероятность того, что одно из этих условий не выполняется. Заметим, что если для некоторого  $r$

$$\Pr_{x,b}[\mathcal{B}(F(x, \hat{u}(x), r)) = 1] - \Pr_{x,b}[\mathcal{B}(F(x, b, r)) = 1] \geq \frac{1}{4m},$$

то  $r$  даёт  $(\frac{1}{2} + \frac{1}{4m})$ -приближение к  $\hat{u}$ . Назовём  $r$  плохим, если оно не даёт  $(\frac{1}{2} + \frac{1}{4m})$ -приближения к  $\hat{u}$ . Из уравнения (7) и неравенства Маркова следует, что

$$\Pr_r[r \text{ плохое}] \leq \frac{1 - 1/2m}{1 - 1/4m} < 1 - \frac{1}{4m}.$$

Согласно неравенству Чернова, для некоторой константы  $c_1 > 0$  выполнено

$$\Pr_{r_1, \dots, r_s} \left[ \text{плохих больше, чем } (1 - \frac{1}{8m})s \right] \leq \exp(-c_1 s/m^2).$$

Для второго условия, также по неравенству Чернова получаем для некоторой константы  $c_2$

$$\Pr \left[ \left| \frac{1}{s} \sum_{j=1}^s w(z_j) - \bar{a} \right| \geq \gamma \right] \leq 2 \exp(-c_2 \gamma^2 s/\bar{n}^2).$$

Взяв  $s = c_3 \bar{n}^2/\gamma^2$  для достаточно большой константы  $c^3$ , получаем верхнюю оценку  $1/8$  в каждом случае, откуда следует исходное утверждение.  $\square$

После выбора слов  $r_1, \dots, r_s$  Артур просит у Мерлина  $s\bar{a} - s\gamma$  сертификатов принадлежности различных слов к  $\mathcal{B}$  и проверяет их. Если хоть один из них ложный, то вычисления заканчиваются без возвращения ответа. Иначе Артур вычисляет строки  $z'_{r_1}, \dots, z'_{r_s}$ , где  $x$ -ый бит строки  $z'_{r_j}$  равен 1 тогда и только тогда, когда Мерлин предоставил сертификат того, что  $\mathcal{B}(F(x, 1, r_j)) = 1$ . Покажем, что с высокой вероятностью вне зависимости от предоставленных Мерлином сертификатов доля  $\frac{1}{16m}$  строк  $z'_{r_1}, \dots, z'_{r_s}$  дадут  $(\frac{1}{2} + \frac{1}{8m})$ -приближение к  $\hat{u}$ .

**Утверждение 8.16.** *Если  $r_1, \dots, r_s$  удовлетворяют условиям предыдущего утверждения для  $\gamma = \bar{n}/256m^2$ , то вне зависимости от предоставленных Мерлином сертификатов доля  $\frac{1}{16m}$  строк  $z'_{r_1}, \dots, z'_{r_s}$  дадут  $(\frac{1}{2} + \frac{1}{8m})$ -приближение к  $\hat{u}$ .*

*Доказательство.* По предположению, число положительных значений  $\mathcal{B}$  для  $r_1, \dots, r_s$  лежит между  $s\bar{a} - s\gamma$  и  $s\bar{a} + s\gamma$ , при этом  $s\bar{a} - s\gamma$  из них известны Артуру. Поскольку Артур проверил переданные сертификаты, то в тех позициях, где в  $z'_{r_j}$  стоит 1, в  $z_{r_j}$  тоже стоит 1. Общее количество различий  $z'_{r_j}$  от  $z_{r_j}$  не превосходит  $2s\gamma$ , поэтому число таких  $r_j$ , где строки различаются в  $t$  битах, не больше  $2s\gamma/t$ .

По предположению, доля  $1/8m$  строк  $z_{r_j}$  дадут  $(\frac{1}{2} + \frac{1}{4m})$ -приближение к  $\hat{u}$ . Положив  $t = \bar{n}/8m$  и  $\gamma = \bar{n}/256m^2$ , получим, что по крайней мере доля  $\frac{1}{8m} - \frac{2\gamma}{t} = \frac{1}{16m}$  строк  $z'_{r_j}$  совпадают с  $\hat{u}$  не меньше чем на доле  $\frac{1}{2} + \frac{1}{4m} - \frac{1}{8m}$  битов.  $\square$

Соединив эти утверждения для достаточно большого полинома  $s$ , например,  $s = \omega(m^4)$ , получаем, что с вероятностью по крайней мере  $3/4$  по меньшей мере доля  $\frac{1}{16m}$  строк  $r_1, \dots, r_s$  дадут  $(\frac{1}{2} + \frac{1}{8m})$ -приближение к  $\hat{u}$ . Всякое конкретное  $r_j$  дадут  $(\frac{1}{2} + \frac{1}{8m})$ -приближение лишь для тех кодовых слов, которые совпадают с  $z_{r_j}$  по крайней мере на доле  $\frac{1}{2} + \frac{1}{8m}$  позиций. По свойству кода таких слов не больше некоторого полинома  $q(m)$ .

Назовем  $\hat{v}$  кандидатом ( $\hat{v} \in \text{Cand}$ ), если не меньше доли  $1/32m$  от всех  $r \in \{0, 1\}^{m-i}$  дают  $(\frac{1}{2} + \frac{1}{8m})$ -приближение для  $\hat{v}$ . Всего кандидатов не больше, чем  $32mq$ , поскольку число рчбер, соединяющих кандидаты и  $r_j$ , не больше  $2^{m-i}q$  и не меньше  $|\text{Cand}| \cdot 2^{m-i}/32m$ . По теореме 8.3 существует программа  $p_1$  длины не больше  $2 \log(32mq) = O(\log n)$ , принимающая  $\hat{u}$  и отвергающая все остальные кандидаты  $\hat{v}$ .

Построим список всех кодовых слов  $\hat{v}$ , совпадающих с одним из  $z'_{r_1}, \dots, z'_{r_s}$  хотя бы на доле  $\frac{1}{2} + \frac{1}{8m}$  от всех позиций. Затем удалим все слова, встретившиеся меньше  $s/16m$  раз. С вероятностью, превосходящей  $2/3$ ,  $\hat{u}$  осталось в списке, и все оставшиеся в списке слова являются кандидатами. В таком случае из всех оставшихся элементов списка  $p_1$  примет  $\hat{u}$  и только его. Поскольку список получен за полиномиальное время, нам не нужно обращаться к оракулу.

Таким образом, мы получили  $\hat{u}$ , зная  $\hat{u}_1, \dots, \hat{u}_{i-1}$  и следующую дополнительную информацию: номер индекса  $i$ , бит  $b_1$ , среднее число  $\bar{a}$  положительных значений  $\mathcal{B}$  и различающую программу  $p_1$ . Заметим, что нам достаточно знать приближение к  $\bar{a}$ , заданное  $O(\log n)$  битами (так чтобы была определена целая часть  $s\bar{a}$ ), поэтому вся дополнительная информация занимает  $O(\log n)$  битов.  $\square$

*Замечание.* Теорема была сформулирована для длины  $X$ , не превосходящей  $k + O(\log^3 n)$  и сложности  $SAM^q(A|B, X) \leq O(\log n)$ . Разумеется, можно «перекинуть» лишние биты из  $X$  в программу алгоритма АМ и получить теорему для  $X$  длины  $k$  и  $SAM^q(A|B, X) \leq O(\log^3 n)$ .

## 8.6. Предмет дальнейших исследований

Основным вопросом теории экстракторов остается построение полиномиально вычислимого оптимального экстрактора. Его решение позволит (помимо прочих замечательных приложений) получить эффективный вариант теоремы Мучника с исходной точностью. Более слабой задачей является построение оптимального экстрактора, вычислимого на полиномиальной зоне. Насколько известно автору, эта задача не решена и даже не исследовалась. Поскольку в доказательстве теоремы Мучника использовалось не свойство экстрактора, а некоторое его следствие, возможно, есть способ построить «псевдоэкстракторы» с оптимальными параметрами, для которых выполнено это следствие, но не основное свойство. Также остается вопрос, верна ли теорема Мучника для полиномиального ограничения на время в формулировках, отличных от теоремы 8.14.

## 9. Благодарности

Автор благодарен А.Е.Ромашенко и А.Шеню за постановку задачи, плодотворное обсуждение и конструктивную критику, а также А.Ю.Румянцеву за обсуждение результата для сложности с ограничением на память. Кроме того, автор благодарен оргкомитету Турнира Городов и лично С.А.Дориченко за включение в состав варианта осеннего тура 2005 года задачи, составленной по мотивам утверждения 2.1.

## Литература

- [1] N.Alon, O.Goldreich, J.Håstad, R.Peralta, "Simple constructions of almost  $k$ -wise independent random variables", Random Structures and Algorithms, 3(3): 289–303, 1992.
- [2] H.Buhrman, L.Fortnow, S.Laplante, Resource bounded Kolmogorov complexity revisited, SIAM Journal on Computing, 31(3):887–905, 2002.
- [3] H.Buhrman, T.Lee, D.van Melkebeek, Language Compression and Pseudorandom Generators, In 19th Annual IEEE Conference on Computational Complexity, IEEE, 2004.
- [4] R.Impagliazzo, L.Levin, M.Luby, "Pseudo-random generation from one-way functions", Proceedings, 21st Annual ACM Symposium on the Theory of Computing, ACM, 1989, 12–24

- [5] M. Li, P. Vitányi, "An introduction to Kolmogorov complexity and its applications", 2nd Edition, Springer, 1997.
- [6] An. Muchnik, Conditional complexity and codes, Theoretical Computer Science, v. 271, issues 1–2, 97–109, 2002.
- [7] J. Naor, M. Naor, "Small-bias probability spaces: efficient constructions and applications", SIAM J. Comput., 22(4): 838–856, 1993.
- [8] N. Nisan, A. Ta-Shma, "Extracting randomness: A survey and new constructions", Journal of Computer and System Sciences, 58(1): 148–173, 1999.
- [9] N. Nisan, A. Wigderson, "Hardness vs. randomness", Journal of Computer and System Sciences, 49: 149–167, 1994.
- [10] N. Nisan, D. Zuckerman, "More deterministic simulation in logspace", Proceedings, 25th Annual ACM Symposium on the Theory of Computing, ACM, 1993, 235–244
- [11] J. Radhakrishnan, A. Ta-Shma, "Bounds for dispersers, extractors, and depth-two superconcentrators", SIAM Journal on Discrete Mathematics, 13(1): 2–24, 2000
- [12] R. Raz, O. Reingold, S. Vadhan, "Extracting all the randomness and reducing the error in Trevisan's extractor", Proceedings, 30th Annual ACM Symposium on the Theory of Computing, ACM, 1999, 149–158.
- [13] O. Reingold, R. Shaltiel, A. Wigderson, "Extracting randomness via repeated condensing", SIAM journal on computing 35(5):1185–1209, 2006.
- [14] R. Shaltiel, "Recent developments in explicit constructions of extractors"
- [15] A. Srinivasan, D. Zuckerman. "Computing with very weak random sources", Proceedings, 35th Annual IEEE Symposium of the Foundation on Computer Science: 264–275, IEEE, 1994.
- [16] L. Trevisan, "Construction of extractors using pseudo-random generators", Journal of the ACM, 48(4):860–879, 2001.